

ISO 10303-210

Product data representation and exchange: Application protocol: Electronic assembly, interconnect, and packaging design

COPYRIGHT NOTICE:

ABSTRACT:

This document specifies the use of ISO 10303 integrated resources that satisfy the information sharing requirements for design engineers, device design engineers, value added engineering data suppliers, design analysts, system engineers, and manufacturing process planning engineers. The information is shared to transform the detailed design of and requirements for either an electronic assembly or an interconnect substrate into either a manufacturable description or an analysable description. Device data is shared to assure adequate product design and for source control purposes.

KEYWORDS: Component Data, Electronic Assembly, Tolerance, Requirements, Interconnect Substrate.

COMMENTS TO READER:

This is an internal committee working draft. The ARM EXPRESS, mapping tables, and AIM structure have been tested. The documentation is incomplete and may not match the EXPRESS. The EXPRESS G for both the ARM and AIM is to be ignored. This work has been done under the sponsorship of PDES, Inc. by the Electrical/Electronics project.

Project Leader: Thomas R. Thurman
Address: Rockwell Collins Inc.
MS 106-183
400 Collins Rd. N.E
Cedar Rapids, IA 52498 U.S.A.

Telephone: 319.295.2280
Telefacsimile: 319.295.0649

Electronic mail: trthurma@rockwellcollins.com

Project Editor: Darla Nettles
Address: Arthur D. Little, Inc.
5300 International Blvd
N. Charleston, SC 29418 U.S.A.

Telephone: 843.760.3232
Telefacsimile: 843.760.3349

Electronic mail: nettles@aticorp.org

Contents	Page
1 Scope	1
2 Normative references	3
3 Terms, definitions, and abbreviations	6
3.1 Terms defined in IEC 60050-541	6
3.2 Terms defined in ISO 129	6
3.3 Terms defined in ISO 1101	6
3.4 Terms defined in ISO 2692	7
3.5 Terms defined in ISO 2692: Amendment 1	7
3.6 Terms defined in ISO 5459	7
3.7 Terms defined in ISO 8015	7
3.8 Terms defined in ISO 10303-1	7
3.9 Terms defined in ISO 10303-11	8
3.10 Terms defined in ISO 10303-44	8
3.11 Terms defined in ISO 10303-45	9
3.12 Terms defined in ISO 10303-46	9
3.13 Other definitions	9
3.14 Abbreviations	11
4 Information requirements	13
4.1 Units of functionality	13
4.2 Application objects	96
4.3 Application assertions	1150
5 Application interpreted model	1363
5.1 Mapping table	1363
5.2 AIM EXPRESS short listing	2933
6 Conformance requirements	3635
6.1 Device Functional and Physical Characterization	3638
6.2 Interconnect Technology Constraints	3640
6.3 Assembly Technology Constraints	3641
6.4 Assembly Functional Requirements	3643
6.5 Assembly Physical Requirements	3654
6.6 Interconnect Functional Requirements	3655
6.7 Interconnect Physical Requirements	3666
6.8 Assembly Physical Design	3667
6.9 Interconnect Design	3681
6.10 Interconnect Design (Microwave)	3698
6.11 Geometric Dimensioning and Tolerancing	3715
6.12 Product Rule	3715
6.13 Functional Decomposition	3721
6.14 Package Functional and Physical Characterization	3729
6.15 Geometrically Bounded Surface Model	3730

6.16	Wireframe Model With Topology	3731
6.17	Advanced Boundary Representation	3732
6.18	Constructive Solid Geometry	3732
6.19	Extruded Solid	3733
6.20	Geometrically Bounded 2d Wireframe Model	3734
6.21	Wireframe 2d Model With Topology	3734
6.22	Curve 2d	3734
6.23	Basic Curve 2d	3735
6.24	Laminate Assembly Design	3735
6.25	Connection Zone Based Model Extraction	3742
6.26	Functional Specification	3747
6.27	Physical Unit Physical Characterization	3752
6.28	Packaged Part White Box Model	3754
6.29	Printed Part Functional and Physical Characterization	3756
6.30	Open Shell Model	3757
6.31	Assembly Functional Interface Requirements	3757
6.32	Assembly Physical Interface Requirements	3765
6.33	Assembly Requirement Allocation	3773
6.34	Interconnect Requirement Allocation	3782
6.35	Functional Requirement Allocation	3791
6.36	Product Requirement Allocation	3794
6.37	Datum Difference Based Model Definition	3797
6.38	Datum Difference Based Characteristic	3800
6.39	Design Management	3804
Annex A (normative)	AIM EXPRESS expanded listing	3810
Annex B (normative)	Short names of entities	4301
Annex C (normative)	Implementation method specific requirements	4318
Annex D (normative)	Protocol Implementation Conformance Statement (PICS) proforma . .	4319
Annex E (normative)	Information object registration	4322
E.1	Document identification	4322
E.2	Schema identification	4322
Annex F (normative)	Application interpreted constructs	4323
F.1	Geometrically bounded surface	4323
F.2	Manifold surface	4341
Annex G (informative)	Application activity model	4358
G.1	Application activity model definitions	4358
G.2	Application activity model figures	4378
Annex H (informative)	Application reference model	4400
H.1	EXPRESS	4400
H.2	EXPRESS-G	4614

Annex J (informative)	AIM EXPRESS-G	4858
Annex K (informative)	AIM EXPRESS listing	4939
Annex L (informative)	Technical discussions	4940
Index		5003

Figures

1	Data planning model	vii
2	Altered_package	104
3	Analytical_model of a resistor	111
4	Array_placement_group	129
5	Assembled_with_bonding	130
6	Assembled_with_fasteners	132
7	Assembly_component	137
8	Assembly_joint	145
9	Assembly_module_usage_view	159
10	Bevel_edge_feature_shape	175
11	Blind_via	178
12	Fabrication drawing detail	187
13	Alternative functions for Bounding_Curve	188
14	Buried_via	190
15	Component_termination_passage role in an assembly	255
16	Component_termination_passage cross-section view	255
17	Component_termination_passage_template terminals	265
18	Partially exploded view of a Conductor	285
19	Conductor	293
20	Contact_size_dependent_land	312
21	Square corners with extend	321
22	Corners with positive extend	322
23	Square corners with truncate	322
24	Round corners	323
25	Curve end treatment	323
26	Completed curve	324
27	Cutout plan view showing installed component	329
28	Cutout cross-section view showing installed component	329
29	Cutout_edge_segment	331
30	Datum	337
31	Datum_based_length_measure	343
32	Datum_system	361
33	Measurement arrangement for minimum_annular_ring	377
34	Dependently_located_layer_connection_point	384
35	Power Plane 1 design intent	393
36	Power Plane 2 design intent	393

37	Signal 3 design intent	393
38	Power Plane 1 to 2 Material removal template	394
39	Power Plane 1 to Signal 3 Material removal template	394
40	Power Plane 2 to Signal 3 Material removal template	394
41	Final shape for the combined signals	395
42	Dielectric_material_passage exploded view	417
43	Electrical_isolation_removal_template supports electrical isolation	479
44	Electrical_isolation_removal_template supports thermal isolation and electrical conduction	480
45	Explicitly_located_layer_connection_point	492
46	Land to Conductive_interconnect_element Fabrication_joint exploded view	498
47	Land to Component_termination_passage exploded view	499
48	Schematic presentation of network definition with Functional_units	526
49	Complex technology specific land pattern definition	566
50	Simple technology specific land pattern definition	567
51	Interconnect_module_component_terminal in an assembly	573
52	Interconnect_module_terminal	592
53	Interface_component	598
54	Interfacial_connection	603
55	Laminate_text_component	617
56	Land	621
57	Custom designs for Land_physical_template	625
58	Complex land template shapes	626
59	Land_template_interface_terminal	628
60	Land_template_join_terminal	630
61	Material_removal_feature_template supports electrical isolation	687
62	Material_removal_feature_template supports thermal isolation and electrical conduction	688
63	Next_higher_assembly_relationship	707
64	Axial lead	727
65	Eight lead SOIC	728
66	Eight lead DIP	729
67	Twenty lead reversed J lead	730
68	Ball grid array	731
69	T099	733
70	Package_terminal	745
71	Packaged_component_join_terminal	752
72	Packaged_connector	754
73	Packaged_connector_component	756
74	Packaged_part exploded assembly detail	760
75	Pca	817
76	Pca_usage_view	819
77	Circular Pcb tolerances	821
78	Complex Pcb tolerances	822
79	Bare printed wiring board	823
80	Physical_unit_design_view	851
81	Plated_inter_stratum_feature cross-section	873
82	Exploded view of Plated_inter_stratum_feature	874

83	Plated_interconnect_module_edge_segment	875
84	Plated_passage	877
85	Three-dimensional assembly relationships	899
86	Two-dimensional assembly relationships	900
87	Twenty pin orientation features	901
88	Eight lead gull wing orientation features	902
89	Eight lead DIP orientation features	903
90	TO-99 orientation features	904
91	Ball grid array orientation features	905
92	Axial Package modifications and resulting orientations	906
93	Primary_reference_terminal and Polarity_indication_feature for discrete devices	906
94	Printed_connector_component and associated terminals	913
95	Single contact of a Printed_connector_template	915
96	Printed_part_cross_section_template	919
97	Printed_part_template	923
98	Probe_access_area	936
99	Process_specification	937
100	Radius_edge_feature_shape	956
101	Shield	1007
102	Step_edge_feature_shape	1026
103	Stratum	1029
104	Stratum_feature	1036
105	Styled_text	1062
106	Sub_assembly_relationship	1064
107	Thermal_isolation_removal_structure	1086
108	Thermal_isolation_removal_template supports electrical isolation	1087
109	Thermal_isolation_removal_template supports thermal isolation and electrical conduction	1088
110	Trace_template	1113
111	Via	1127
112	Via_template_terminal	1134
G.1	Electrical or electronic product life cycle (Context)	4379
G.2	Create electrical-electronic product	4380
G.3	Develop product	4381
G.4	Develop preliminary product definition	4382
G.5	Develop detailed architecture	4383
G.6	Complete detailed architecture	4384
G.7	Define electronic design	4385
G.8	Analyse electronic design	4386
G.9	Analyse board specialty disciplines	4387
G.10	Develop detail product definition	4388
G.11	Design mechanical PCA	4389
G.12	Evaluate functional mechanical characteristics	4390
G.13	Finish detailed PCA definition	4391
G.14	Place components	4392
G.15	Locate components	4393
G.16	Refine placement	4394

G.17	Analyse component placement	4395
G.18	Route board	4396
G.19	Prepare for routing	4397
G.20	Make prototype	4398
G.21	Release engineering design	4399
H.1	ARM EXPRESS-G Graphic 1 of 243	4615
H.2	ARM EXPRESS-G Graphic 2 of 243	4616
H.3	ARM EXPRESS-G Graphic 3 of 243	4617
H.4	ARM EXPRESS-G Graphic 4 of 243	4618
H.5	ARM EXPRESS-G Graphic 5 of 243	4619
H.6	ARM EXPRESS-G Graphic 6 of 243	4620
H.7	ARM EXPRESS-G Graphic 7 of 243	4621
H.8	ARM EXPRESS-G Graphic 8 of 243	4622
H.9	ARM EXPRESS-G Graphic 9 of 243	4623
H.10	ARM EXPRESS-G Graphic 10 of 243	4624
H.11	ARM EXPRESS-G Graphic 11 of 243	4625
H.12	ARM EXPRESS-G Graphic 12 of 243	4626
H.13	ARM EXPRESS-G Graphic 13 of 243	4627
H.14	ARM EXPRESS-G Graphic 14 of 243	4628
H.15	ARM EXPRESS-G Graphic 15 of 243	4629
H.16	ARM EXPRESS-G Graphic 16 of 243	4630
H.17	ARM EXPRESS-G Graphic 17 of 243	4631
H.18	ARM EXPRESS-G Graphic 18 of 243	4632
H.19	ARM EXPRESS-G Graphic 19 of 243	4633
H.20	ARM EXPRESS-G Graphic 20 of 243	4634
H.21	ARM EXPRESS-G Graphic 21 of 243	4635
H.22	ARM EXPRESS-G Graphic 22 of 243	4636
H.23	ARM EXPRESS-G Graphic 23 of 243	4637
H.24	ARM EXPRESS-G Graphic 24 of 243	4638
H.25	ARM EXPRESS-G Graphic 25 of 243	4639
H.26	ARM EXPRESS-G Graphic 26 of 243	4640
H.27	ARM EXPRESS-G Graphic 27 of 243	4641
H.28	ARM EXPRESS-G Graphic 28 of 243	4642
H.29	ARM EXPRESS-G Graphic 29 of 243	4643
H.30	ARM EXPRESS-G Graphic 30 of 243	4644
H.31	ARM EXPRESS-G Graphic 31 of 243	4645
H.32	ARM EXPRESS-G Graphic 32 of 243	4646
H.33	ARM EXPRESS-G Graphic 33 of 243	4647
H.34	ARM EXPRESS-G Graphic 34 of 243	4648
H.35	ARM EXPRESS-G Graphic 35 of 243	4649
H.36	ARM EXPRESS-G Graphic 36 of 243	4650
H.37	ARM EXPRESS-G Graphic 37 of 243	4651
H.38	ARM EXPRESS-G Graphic 38 of 243	4652
H.39	ARM EXPRESS-G Graphic 39 of 243	4653
H.40	ARM EXPRESS-G Graphic 40 of 243	4654
H.41	ARM EXPRESS-G Graphic 41 of 243	4655

H.42	ARM EXPRESS-G Graphic 42 of 243	4656
H.43	ARM EXPRESS-G Graphic 43 of 243	4657
H.44	ARM EXPRESS-G Graphic 44 of 243	4658
H.45	ARM EXPRESS-G Graphic 45 of 243	4659
H.46	ARM EXPRESS-G Graphic 46 of 243	4660
H.47	ARM EXPRESS-G Graphic 47 of 243	4661
H.48	ARM EXPRESS-G Graphic 48 of 243	4662
H.49	ARM EXPRESS-G Graphic 49 of 243	4663
H.50	ARM EXPRESS-G Graphic 50 of 243	4664
H.51	ARM EXPRESS-G Graphic 51 of 243	4665
H.52	ARM EXPRESS-G Graphic 52 of 243	4666
H.53	ARM EXPRESS-G Graphic 53 of 243	4667
H.54	ARM EXPRESS-G Graphic 54 of 243	4668
H.55	ARM EXPRESS-G Graphic 55 of 243	4669
H.56	ARM EXPRESS-G Graphic 56 of 243	4670
H.57	ARM EXPRESS-G Graphic 57 of 243	4671
H.58	ARM EXPRESS-G Graphic 58 of 243	4672
H.59	ARM EXPRESS-G Graphic 59 of 243	4673
H.60	ARM EXPRESS-G Graphic 60 of 243	4674
H.61	ARM EXPRESS-G Graphic 61 of 243	4675
H.62	ARM EXPRESS-G Graphic 62 of 243	4676
H.63	ARM EXPRESS-G Graphic 63 of 243	4677
H.64	ARM EXPRESS-G Graphic 64 of 243	4678
H.65	ARM EXPRESS-G Graphic 65 of 243	4679
H.66	ARM EXPRESS-G Graphic 66 of 243	4680
H.67	ARM EXPRESS-G Graphic 67 of 243	4681
H.68	ARM EXPRESS-G Graphic 68 of 243	4682
H.69	ARM EXPRESS-G Graphic 69 of 243	4683
H.70	ARM EXPRESS-G Graphic 70 of 243	4684
H.71	ARM EXPRESS-G Graphic 71 of 243	4685
H.72	ARM EXPRESS-G Graphic 72 of 243	4686
H.73	ARM EXPRESS-G Graphic 73 of 243	4687
H.74	ARM EXPRESS-G Graphic 74 of 243	4688
H.75	ARM EXPRESS-G Graphic 75 of 243	4689
H.76	ARM EXPRESS-G Graphic 76 of 243	4690
H.77	ARM EXPRESS-G Graphic 77 of 243	4691
H.78	ARM EXPRESS-G Graphic 78 of 243	4692
H.79	ARM EXPRESS-G Graphic 79 of 243	4693
H.80	ARM EXPRESS-G Graphic 80 of 243	4694
H.81	ARM EXPRESS-G Graphic 81 of 243	4695
H.82	ARM EXPRESS-G Graphic 82 of 243	4696
H.83	ARM EXPRESS-G Graphic 83 of 243	4697
H.84	ARM EXPRESS-G Graphic 84 of 243	4698
H.85	ARM EXPRESS-G Graphic 85 of 243	4699
H.86	ARM EXPRESS-G Graphic 86 of 243	4700
H.87	ARM EXPRESS-G Graphic 87 of 243	4701

H.88	ARM EXPRESS-G Graphic 88 of 243	4702
H.89	ARM EXPRESS-G Graphic 89 of 243	4703
H.90	ARM EXPRESS-G Graphic 90 of 243	4704
H.91	ARM EXPRESS-G Graphic 91 of 243	4705
H.92	ARM EXPRESS-G Graphic 92 of 243	4706
H.93	ARM EXPRESS-G Graphic 93 of 243	4707
H.94	ARM EXPRESS-G Graphic 94 of 243	4708
H.95	ARM EXPRESS-G Graphic 95 of 243	4709
H.96	ARM EXPRESS-G Graphic 96 of 243	4710
H.97	ARM EXPRESS-G Graphic 97 of 243	4711
H.98	ARM EXPRESS-G Graphic 98 of 243	4712
H.99	ARM EXPRESS-G Graphic 99 of 243	4713
H.100	ARM EXPRESS-G Graphic 100 of 243	4714
H.101	ARM EXPRESS-G Graphic 101 of 243	4715
H.102	ARM EXPRESS-G Graphic 102 of 243	4716
H.103	ARM EXPRESS-G Graphic 103 of 243	4717
H.104	ARM EXPRESS-G Graphic 104 of 243	4718
H.105	ARM EXPRESS-G Graphic 105 of 243	4719
H.106	ARM EXPRESS-G Graphic 106 of 243	4720
H.107	ARM EXPRESS-G Graphic 107 of 243	4721
H.108	ARM EXPRESS-G Graphic 108 of 243	4722
H.109	ARM EXPRESS-G Graphic 109 of 243	4723
H.110	ARM EXPRESS-G Graphic 110 of 243	4724
H.111	ARM EXPRESS-G Graphic 111 of 243	4725
H.112	ARM EXPRESS-G Graphic 112 of 243	4726
H.113	ARM EXPRESS-G Graphic 113 of 243	4727
H.114	ARM EXPRESS-G Graphic 114 of 243	4728
H.115	ARM EXPRESS-G Graphic 115 of 243	4729
H.116	ARM EXPRESS-G Graphic 116 of 243	4730
H.117	ARM EXPRESS-G Graphic 117 of 243	4731
H.118	ARM EXPRESS-G Graphic 118 of 243	4732
H.119	ARM EXPRESS-G Graphic 119 of 243	4733
H.120	ARM EXPRESS-G Graphic 120 of 243	4734
H.121	ARM EXPRESS-G Graphic 121 of 243	4735
H.122	ARM EXPRESS-G Graphic 122 of 243	4736
H.123	ARM EXPRESS-G Graphic 123 of 243	4737
H.124	ARM EXPRESS-G Graphic 124 of 243	4738
H.125	ARM EXPRESS-G Graphic 125 of 243	4739
H.126	ARM EXPRESS-G Graphic 126 of 243	4740
H.127	ARM EXPRESS-G Graphic 127 of 243	4741
H.128	ARM EXPRESS-G Graphic 128 of 243	4742
H.129	ARM EXPRESS-G Graphic 129 of 243	4743
H.130	ARM EXPRESS-G Graphic 130 of 243	4744
H.131	ARM EXPRESS-G Graphic 131 of 243	4745
H.132	ARM EXPRESS-G Graphic 132 of 243	4746
H.133	ARM EXPRESS-G Graphic 133 of 243	4747

H.134	ARM EXPRESS-G Graphic 134 of 243	4748
H.135	ARM EXPRESS-G Graphic 135 of 243	4749
H.136	ARM EXPRESS-G Graphic 136 of 243	4750
H.137	ARM EXPRESS-G Graphic 137 of 243	4751
H.138	ARM EXPRESS-G Graphic 138 of 243	4752
H.139	ARM EXPRESS-G Graphic 139 of 243	4753
H.140	ARM EXPRESS-G Graphic 140 of 243	4754
H.141	ARM EXPRESS-G Graphic 141 of 243	4755
H.142	ARM EXPRESS-G Graphic 142 of 243	4756
H.143	ARM EXPRESS-G Graphic 143 of 243	4757
H.144	ARM EXPRESS-G Graphic 144 of 243	4758
H.145	ARM EXPRESS-G Graphic 145 of 243	4759
H.146	ARM EXPRESS-G Graphic 146 of 243	4760
H.147	ARM EXPRESS-G Graphic 147 of 243	4761
H.148	ARM EXPRESS-G Graphic 148 of 243	4762
H.149	ARM EXPRESS-G Graphic 149 of 243	4763
H.150	ARM EXPRESS-G Graphic 150 of 243	4764
H.151	ARM EXPRESS-G Graphic 151 of 243	4765
H.152	ARM EXPRESS-G Graphic 152 of 243	4766
H.153	ARM EXPRESS-G Graphic 153 of 243	4767
H.154	ARM EXPRESS-G Graphic 154 of 243	4768
H.155	ARM EXPRESS-G Graphic 155 of 243	4769
H.156	ARM EXPRESS-G Graphic 156 of 243	4770
H.157	ARM EXPRESS-G Graphic 157 of 243	4771
H.158	ARM EXPRESS-G Graphic 158 of 243	4772
H.159	ARM EXPRESS-G Graphic 159 of 243	4773
H.160	ARM EXPRESS-G Graphic 160 of 243	4774
H.161	ARM EXPRESS-G Graphic 161 of 243	4775
H.162	ARM EXPRESS-G Graphic 162 of 243	4776
H.163	ARM EXPRESS-G Graphic 163 of 243	4777
H.164	ARM EXPRESS-G Graphic 164 of 243	4778
H.165	ARM EXPRESS-G Graphic 165 of 243	4779
H.166	ARM EXPRESS-G Graphic 166 of 243	4780
H.167	ARM EXPRESS-G Graphic 167 of 243	4781
H.168	ARM EXPRESS-G Graphic 168 of 243	4782
H.169	ARM EXPRESS-G Graphic 169 of 243	4783
H.170	ARM EXPRESS-G Graphic 170 of 243	4784
H.171	ARM EXPRESS-G Graphic 171 of 243	4785
H.172	ARM EXPRESS-G Graphic 172 of 243	4786
H.173	ARM EXPRESS-G Graphic 173 of 243	4787
H.174	ARM EXPRESS-G Graphic 174 of 243	4788
H.175	ARM EXPRESS-G Graphic 175 of 243	4789
H.176	ARM EXPRESS-G Graphic 176 of 243	4790
H.177	ARM EXPRESS-G Graphic 177 of 243	4791
H.178	ARM EXPRESS-G Graphic 178 of 243	4792
H.179	ARM EXPRESS-G Graphic 179 of 243	4793

H.180	ARM EXPRESS-G Graphic 180 of 243	4794
H.181	ARM EXPRESS-G Graphic 181 of 243	4795
H.182	ARM EXPRESS-G Graphic 182 of 243	4796
H.183	ARM EXPRESS-G Graphic 183 of 243	4797
H.184	ARM EXPRESS-G Graphic 184 of 243	4798
H.185	ARM EXPRESS-G Graphic 185 of 243	4799
H.186	ARM EXPRESS-G Graphic 186 of 243	4800
H.187	ARM EXPRESS-G Graphic 187 of 243	4801
H.188	ARM EXPRESS-G Graphic 188 of 243	4802
H.189	ARM EXPRESS-G Graphic 189 of 243	4803
H.190	ARM EXPRESS-G Graphic 190 of 243	4804
H.191	ARM EXPRESS-G Graphic 191 of 243	4805
H.192	ARM EXPRESS-G Graphic 192 of 243	4806
H.193	ARM EXPRESS-G Graphic 193 of 243	4807
H.194	ARM EXPRESS-G Graphic 194 of 243	4808
H.195	ARM EXPRESS-G Graphic 195 of 243	4809
H.196	ARM EXPRESS-G Graphic 196 of 243	4810
H.197	ARM EXPRESS-G Graphic 197 of 243	4811
H.198	ARM EXPRESS-G Graphic 198 of 243	4812
H.199	ARM EXPRESS-G Graphic 199 of 243	4813
H.200	ARM EXPRESS-G Graphic 200 of 243	4814
H.201	ARM EXPRESS-G Graphic 201 of 243	4815
H.202	ARM EXPRESS-G Graphic 202 of 243	4816
H.203	ARM EXPRESS-G Graphic 203 of 243	4817
H.204	ARM EXPRESS-G Graphic 204 of 243	4818
H.205	ARM EXPRESS-G Graphic 205 of 243	4819
H.206	ARM EXPRESS-G Graphic 206 of 243	4820
H.207	ARM EXPRESS-G Graphic 207 of 243	4821
H.208	ARM EXPRESS-G Graphic 208 of 243	4822
H.209	ARM EXPRESS-G Graphic 209 of 243	4823
H.210	ARM EXPRESS-G Graphic 210 of 243	4824
H.211	ARM EXPRESS-G Graphic 211 of 243	4825
H.212	ARM EXPRESS-G Graphic 212 of 243	4826
H.213	ARM EXPRESS-G Graphic 213 of 243	4827
H.214	ARM EXPRESS-G Graphic 214 of 243	4828
H.215	ARM EXPRESS-G Graphic 215 of 243	4829
H.216	ARM EXPRESS-G Graphic 216 of 243	4830
H.217	ARM EXPRESS-G Graphic 217 of 243	4831
H.218	ARM EXPRESS-G Graphic 218 of 243	4832
H.219	ARM EXPRESS-G Graphic 219 of 243	4833
H.220	ARM EXPRESS-G Graphic 220 of 243	4834
H.221	ARM EXPRESS-G Graphic 221 of 243	4835
H.222	ARM EXPRESS-G Graphic 222 of 243	4836
H.223	ARM EXPRESS-G Graphic 223 of 243	4837
H.224	ARM EXPRESS-G Graphic 224 of 243	4838
H.225	ARM EXPRESS-G Graphic 225 of 243	4839

H.226	ARM EXPRESS-G Graphic 226 of 243	4840
H.227	ARM EXPRESS-G Graphic 227 of 243	4841
H.228	ARM EXPRESS-G Graphic 228 of 243	4842
H.229	ARM EXPRESS-G Graphic 229 of 243	4843
H.230	ARM EXPRESS-G Graphic 230 of 243	4844
H.231	ARM EXPRESS-G Graphic 231 of 243	4845
H.232	ARM EXPRESS-G Graphic 232 of 243	4846
H.233	ARM EXPRESS-G Graphic 233 of 243	4847
H.234	ARM EXPRESS-G Graphic 234 of 243	4848
H.235	ARM EXPRESS-G Graphic 235 of 243	4849
H.236	ARM EXPRESS-G Graphic 236 of 243	4850
H.237	ARM EXPRESS-G Graphic 237 of 243	4851
H.238	ARM EXPRESS-G Graphic 238 of 243	4852
H.239	ARM EXPRESS-G Graphic 239 of 243	4853
H.240	ARM EXPRESS-G Graphic 240 of 243	4854
H.241	ARM EXPRESS-G Graphic 241 of 243	4855
H.242	ARM EXPRESS-G Graphic 242 of 243	4856
H.243	ARM EXPRESS-G Graphic 243 of 243	4857
J.1	AIM EXPRESS-G diagram 1 of 80	4859
J.2	AIM EXPRESS-G diagram 2 of 80	4860
J.3	AIM EXPRESS-G diagram 3 of 80	4861
J.4	AIM EXPRESS-G diagram 4 of 80	4862
J.5	AIM EXPRESS-G diagram 5 of 80	4863
J.6	AIM EXPRESS-G diagram 6 of 80	4864
J.7	AIM EXPRESS-G diagram 7 of 80	4865
J.8	AIM EXPRESS-G diagram 8 of 80	4866
J.9	AIM EXPRESS-G diagram 9 of 80	4867
J.10	AIM EXPRESS-G diagram 10 of 80	4868
J.11	AIM EXPRESS-G diagram 11 of 80	4869
J.12	AIM EXPRESS-G diagram 12 of 80	4870
J.13	AIM EXPRESS-G diagram 13 of 80	4871
J.14	AIM EXPRESS-G diagram 14 of 80	4872
J.15	AIM EXPRESS-G diagram 15 of 80	4873
J.16	AIM EXPRESS-G diagram 16 of 80	4874
J.17	AIM EXPRESS-G diagram 17 of 80	4875
J.18	AIM EXPRESS-G diagram 18 of 80	4876
J.19	AIM EXPRESS-G diagram 19 of 80	4877
J.20	AIM EXPRESS-G diagram 20 of 80	4878
J.21	AIM EXPRESS-G diagram 21 of 80	4879
J.22	AIM EXPRESS-G diagram 22 of 80	4880
J.23	AIM EXPRESS-G diagram 23 of 80	4881
J.24	AIM EXPRESS-G diagram 24 of 80	4882
J.25	AIM EXPRESS-G diagram 25 of 80	4883
J.26	AIM EXPRESS-G diagram 26 of 80	4884
J.27	AIM EXPRESS-G diagram 27 of 80	4885
J.28	AIM EXPRESS-G diagram 28 of 80	4886

J.29	AIM EXPRESS-G diagram 29 of 80	4887
J.30	AIM EXPRESS-G diagram 30 of 80	4888
J.31	AIM EXPRESS-G diagram 31 of 80	4889
J.32	AIM EXPRESS-G diagram 32 of 80	4890
J.33	AIM EXPRESS-G diagram 33 of 80	4891
J.34	AIM EXPRESS-G diagram 34 of 80	4892
J.35	AIM EXPRESS-G diagram 35 of 80	4893
J.36	AIM EXPRESS-G diagram 36 of 80	4894
J.37	AIM EXPRESS-G diagram 37 of 80	4895
J.38	AIM EXPRESS-G diagram 38 of 80	4896
J.39	AIM EXPRESS-G diagram 39 of 80	4897
J.40	AIM EXPRESS-G diagram 40 of 80	4898
J.41	AIM EXPRESS-G diagram 41 of 80	4899
J.42	AIM EXPRESS-G diagram 42 of 80	4900
J.43	AIM EXPRESS-G diagram 43 of 80	4901
J.44	AIM EXPRESS-G diagram 44 of 80	4902
J.45	AIM EXPRESS-G diagram 45 of 80	4903
J.46	AIM EXPRESS-G diagram 46 of 80	4904
J.47	AIM EXPRESS-G diagram 47 of 80	4905
J.48	AIM EXPRESS-G diagram 48 of 80	4906
J.49	AIM EXPRESS-G diagram 49 of 80	4907
J.50	AIM EXPRESS-G diagram 50 of 80	4908
J.51	AIM EXPRESS-G diagram 51 of 80	4909
J.52	AIM EXPRESS-G diagram 52 of 80	4910
J.53	AIM EXPRESS-G diagram 53 of 80	4911
J.54	AIM EXPRESS-G diagram 54 of 80	4912
J.55	AIM EXPRESS-G diagram 55 of 80	4913
J.56	AIM EXPRESS-G diagram 56 of 80	4914
J.57	AIM EXPRESS-G diagram 57 of 80	4915
J.58	AIM EXPRESS-G diagram 58 of 80	4916
J.59	AIM EXPRESS-G diagram 59 of 80	4917
J.60	AIM EXPRESS-G diagram 60 of 80	4918
J.61	AIM EXPRESS-G diagram 61 of 80	4919
J.62	AIM EXPRESS-G diagram 62 of 80	4920
J.63	AIM EXPRESS-G diagram 63 of 80	4921
J.64	AIM EXPRESS-G diagram 64 of 80	4922
J.65	AIM EXPRESS-G diagram 65 of 80	4923
J.66	AIM EXPRESS-G diagram 66 of 80	4924
J.67	AIM EXPRESS-G diagram 67 of 80	4925
J.68	AIM EXPRESS-G diagram 68 of 80	4926
J.69	AIM EXPRESS-G diagram 69 of 80	4927
J.70	AIM EXPRESS-G diagram 70 of 80	4928
J.71	AIM EXPRESS-G diagram 71 of 80	4929
J.72	AIM EXPRESS-G diagram 72 of 80	4930
J.73	AIM EXPRESS-G diagram 73 of 80	4931
J.74	AIM EXPRESS-G diagram 74 of 80	4932

J.75	AIM EXPRESS-G diagram 75 of 80	4933
J.76	AIM EXPRESS-G diagram 76 of 80	4934
J.77	AIM EXPRESS-G diagram 77 of 80	4935
J.78	AIM EXPRESS-G diagram 78 of 80	4936
J.79	AIM EXPRESS-G diagram 79 of 80	4937
J.80	AIM EXPRESS-G diagram 80 of 80	4938

Tables

1	Mapping table for advanced_boundary_representation UoF	1365
2	Mapping table for altered_package UoF	1366
3	Mapping table for altered_part UoF	1372
4	Mapping table for analytical_model UoF	1386
5	Mapping table for analytical_representation UoF	1399
6	Mapping table for assembly_bill_of_material UoF	1400
7	Mapping table for assembly_component UoF	1411
8	Mapping table for assembly_component_grouping UoF	1429
9	Mapping table for assembly_component_placement_requirements UoF	1446
10	Mapping table for assembly_design_information_assignment_to_usage_view UoF	1470
11	Mapping table for assembly_functional_interface_requirement UoF	1476
12	Mapping table for assembly_functional_requirement_allocation UoF	1479
13	Mapping table for assembly_interface_component UoF	1483
14	Mapping table for assembly_marking UoF	1486
15	Mapping table for assembly_module_design UoF	1492
16	Mapping table for assembly_module_usage_view UoF	1540
17	Mapping table for assembly_physical_interface_requirement UoF	1551
18	Mapping table for assembly_physical_requirement_allocation UoF	1560
19	Mapping table for assembly_shape UoF	1573
20	Mapping table for assembly_technology UoF	1602
21	Mapping table for authorization UoF	1610
22	Mapping table for bare_die UoF	1618
23	Mapping table for characteristic UoF	1635
24	Mapping table for connection_zone_to_analytical_model_port_mapping UoF	1656
25	Mapping table for constructive_solid_geometry UoF	1657
26	Mapping table for datum_based_characteristic UoF	1659
27	Mapping table for datum_difference_based_characteristic UoF	1661
28	Mapping table for datum_difference_based_model UoF	1664
29	Mapping table for design_activity_control UoF	1679
30	Mapping table for design_management UoF	1691
31	Mapping table for device_marking UoF	1702
32	Mapping table for document UoF	1708
33	Mapping table for end_item_identification UoF	1718
34	Mapping table for external_definition UoF	1720
35	Mapping table for fabrication_joint UoF	1722
36	Mapping table for form_features UoF	1729
37	Mapping table for functional_decomposition_to_assembly_design UoF	1741

38	Mapping table for functional_decomposition_to_interconnect_design UoF	1748
39	Mapping table for functional_decomposition_with_nodal_representation UoF	1758
40	Mapping table for functional_decomposition_with_nodal_representation_to_assembly_map- ping UoF	1772
41	Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_- mapping UoF	1786
42	Mapping table for functional_definition_to_physical_unit_usage_view_mapping UoF . . .	1797
43	Mapping table for functional_definition_to_printed_part_mapping UoF	1802
44	Mapping table for functional_design_elaboration UoF	1807
45	Mapping table for functional_requirement_allocation UoF	1816
46	Mapping table for functional_specification UoF	1818
47	Mapping table for functional_unit_analytical_representation UoF	1825
48	Mapping table for functional_usage_view UoF	1827
49	Mapping table for functional_usage_view_terminal_to_analytical_model_port_mapping UoF	1832
50	Mapping table for geometric_dimensioning_and_tolerance UoF	1833
51	Mapping table for geometrically_bounded_2d_wireframe UoF	1941
52	Mapping table for inter_stratum_feature_template UoF	1942
53	Mapping table for interconnect_bill_of_material UoF	1952
54	Mapping table for interconnect_component_grouping UoF	1964
55	Mapping table for interconnect_design_information_assignment_to_usage_view UoF . . .	1988
56	Mapping table for interconnect_functional_requirement_allocation UoF	1998
57	Mapping table for interconnect_interface_component UoF	2005
58	Mapping table for interconnect_marking UoF	2008
59	Mapping table for interconnect_module_connection_routing UoF	2014
60	Mapping table for interconnect_module_design UoF	2046
61	Mapping table for interconnect_module_edge UoF	2212
62	Mapping table for interconnect_module_usage_view UoF	2217
63	Mapping table for interconnect_physical_requirement_allocation UoF	2236
64	Mapping table for interconnect_placement_requirements UoF	2259
65	Mapping table for interconnect_shape UoF	2318
66	Mapping table for laminate_assembly_design UoF	2371
67	Mapping table for land UoF	2382
68	Mapping table for library_item UoF	2404
69	Mapping table for manufacturing_technology UoF	2405
70	Mapping table for material_product UoF	2443
71	Mapping table for non_topological_surface UoF	2449
72	Mapping table for package UoF	2450
73	Mapping table for packaged_part_black_box_model UoF	2496
74	Mapping table for packaged_part_terminal_analytical_representation UoF	2510
75	Mapping table for packaged_part_white_box_model UoF	2513
76	Mapping table for part_feature UoF	2540
77	Mapping table for part_feature_to_analytical_model_port_mapping UoF	2586
78	Mapping table for part_template UoF	2589
79	Mapping table for part_template_shape UoF	2628
80	Mapping table for physical_network UoF	2661
81	Mapping table for physical_unit UoF	2666

82	Mapping table for physical_unit_analytical_representation UoF	2687
83	Mapping table for physical_unit_shape UoF	2691
84	Mapping table for planned_characteristic UoF	2720
85	Mapping table for planned_effectivity UoF	2725
86	Mapping table for printed_part_template UoF	2728
87	Mapping table for product_identification UoF	2760
88	Mapping table for product_requirement_allocation UoF	2788
89	Mapping table for product_rule UoF	2792
90	Mapping table for requirement_decomposition UoF	2804
91	Mapping table for sequential_laminate_assembly_design UoF	2822
92	Mapping table for shape UoF	2833
93	Mapping table for signal UoF	2868
94	Mapping table for solid_of_linear_extrusion UoF	2872
95	Mapping table for source_control UoF	2873
96	Mapping table for stratum UoF	2877
97	Mapping table for stratum_feature UoF	2890
98	Mapping table for stratum_position UoF	2910
99	Mapping table for styled_curve UoF	2921
100	Mapping table for styled_text UoF	2926
101	Mapping table for test_probe UoF	2928
102	Mapping table for wireframe_2d_model_with_topology UoF	2931
103	Mapping table for wireframe_with_topology UoF	2932
B.1	Short names of entities	4301

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

A complete list of parts of ISO 10303 is available from the internet:

<<http://www.nist.gov/sc4/editing/step/titles/>>

International Standard ISO 10303-210 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

Annexes A, B, C, D, E and F form a normative part of this International Standard. Annexes G, H, J, K, and L are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the application protocol series.

This part of ISO 10303 specifies an application protocol (AP) for the representation of information needed for the requirements specification of, design of, and analysis of, electronic assemblies, interconnect, and packaging.

This AP includes information needed to support configuration management of the product data.

The detailed product data identify the functional and physical performance requirements for the product in its intended environment. The functional and physical interface requirements define the usage environment of the product. Configuration management information is part of the product data.

The detailed functional requirements data may be either functional parametric properties data or explicit detailed functional design data. The functional parametric properties data, used in conjunction with an externally defined analysis model, describe the behavior of the product. The detailed functional design data may be used to identify the usage of functional abstractions as components and the connections between those components.

The detailed device data identify the products used in the design and their critical functional and physical characteristics. The detailed device data are from the external or usage view of the device. The functional device characteristics may be represented as detailed functional design data, allowing the association of the detailed functional requirements data to available products.

The analysable description includes information, at a level of abstraction, for physically realizing the product design. The analysable description may include the functional interconnect of descriptions and analysis models. The analysable description may be detailed functional design data that identify connections between components or material features in a design. Sufficient information is included to support multi-level mixed functional or physical analysis of a system composed of the assembly or interconnect products.

The detailed manufacturing capabilities data identify the material properties used as a basis in the design.

The manufacturable description includes information for physically realizing either the electronic assembly or interconnect product design. An assembly or a design may be used as a component in a different design, with geometry providing shape information in a geometric context.

The electronic assembly description is technology independent and identifies the products used as components and provides their placement and geometric shape data. These data support a detailed analysis of the performance of a product in its intended environment.

The interconnect product design describes the materials composing the interconnect, their arrangement in the interconnect physical structure, and their shape. The interconnect describes the arrangement and shape of material features of the interconnect product design. The interconnect description is technology independent.

Figure 1 contains the data planning model that provides a high-level description of the requirements for this application protocol. This planning model was created from the in-scope activities data of the application activity model (AAM). This planning model was used as a guide in developing the application reference model (ARM).

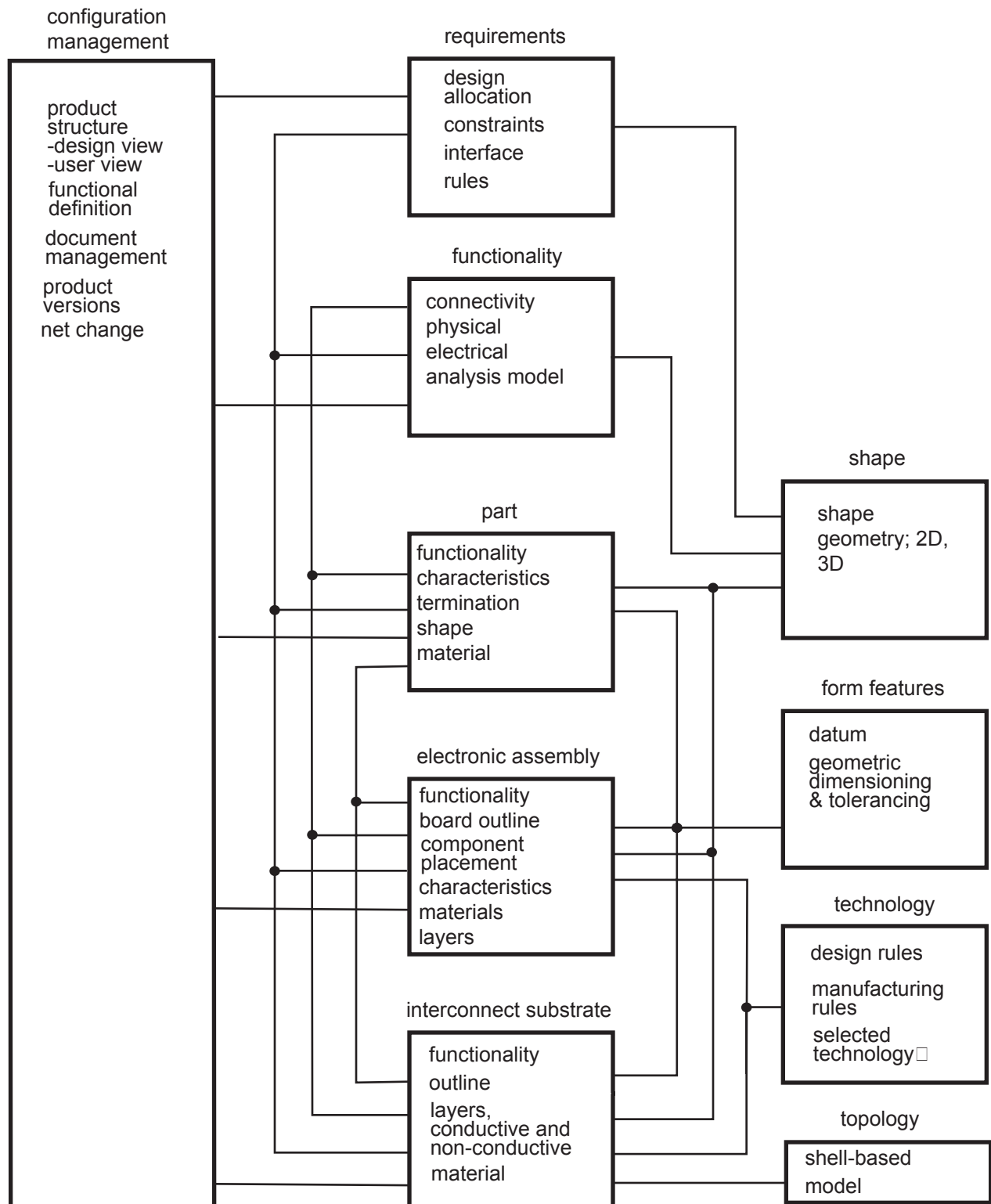


Figure 1 – Data planning model

This application protocol defines the context, scope, and information requirements for information shared between device, system, design, and manufacturing process planning engineers for transforming detailed requirements data, detailed device data, detailed manufacturing capabilities data, and detailed functional design data into analysable and manufacturable descriptions and specifies the integrated resources necessary to satisfy these requirements.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. An application activity model that is the basis for the definition of the scope is provided in annex G. The information requirements for the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex H.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in clause 5.1 shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in annex J. Additional requirements for specific implementation methods are given in annex C.

Industrial automation systems and integration — Product data representation and exchange — Part 210 : Application Protocol: Electronic assembly, interconnect, and packaging design

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the design of electronic assemblies, interconnect and packaging.

NOTE The application activity model (AAM) in Annex G provides a graphical representation of the processes and information flows that are the basis for the definition of the scope of this part of ISO 10303.

The following are within the scope of this part of ISO 10303:

- The hierarchical description of the functionality of the electronic assembly or interconnect or devices that are used in an electronic assembly or interconnect;
- The hierarchical description of the functional objects that are combinations of one or more functional objects;
- The configuration management of the functional objects that are being developed concurrently;
- The configuration management of analytical models that are being developed concurrently;
- The explicit interface to analytic models that are used to define the behaviour of a functional object, a device, a physical object, an electronic assembly, or an interconnect;
- The description of the connection among the functional objects;
- The description of the connection among the devices used in the electronic assembly or interconnect;
- The description of the requirements for physical interconnection;
- The physical layout of the electronic assembly, including a description of the placement of the physical objects in the electronic assembly;
- The description of the bare interconnect, including the conductive and non-conductive materials and patterns;

- The functional and physical description of devices or physical objects, including material characteristics and composition;
- The functional usage and physical usage description of devices or physical objects, including material characteristics and composition;
- The functional and physical description of devices or physical objects realized as part of the interconnect, including material characteristics and composition;
- The functional usage and physical usage description of devices or physical objects realized as part of the interconnect, including material characteristics and composition;
- The description of the requirements and constraints on the design of the electronic assembly that assure product performance, incorporate quality, and enhance manufacturing process capabilities;
- The description of the requirements and constraints on the design of the interconnect that assure product performance, incorporate quality, and enhance manufacturing process capabilities;
- The configuration management of electronic assembly design descriptions;
- The configuration management of interconnect design descriptions;
- The allocation of certain details of the interconnect design description to the usage description of a physical object;
- The allocation of certain details of the electronic assembly design description to the usage description of a physical object;
- The description of electronic assemblies to implement various functional domains including, but not limited to, analogue, digital, video, radio frequency, and microwave;
- The description of interconnect to implement various functional domains including, but not limited to, analogue, digital, video, radio frequency, and microwave;
- The configuration management of constituent parts that are electronic assemblies or interconnect and that are being concurrently developed;
- The configuration management of documents that contain requirements;
- The allocation of requirements to functional objects, physical objects, and the physical implementation;
- The allocation of requirements from functional objects to their physical implementation;
- The allocation of each specific connection among the devices used in the electronic assembly to the component or group of components that realizes that connection;

- The association of characteristics to functional objects, physical objects, and analytical models;
- The identification of planned parameters for functional objects, electronic assemblies, interconnect, components, and analytical models;
- The identification of actual parameters for functional objects, electronic assemblies, interconnect, components, and analytical models.

The following are outside the scope of this part of ISO 10303:

- The drawings of functional objects, physical objects, electronic assemblies and interconnect;
- The process plans for the assembly of the electronic assembly;
- The process plans for the fabrication of the interconnect;
- The classification and categorization of data element types;
- The definition and interpretation of external file formats for analytic models;
- The management of the process used to design an electronic assembly or interconnect;
- The management of the manufacture of the physical objects used in an electronic assembly;
- The administrative procurement and cost data used by an enterprise.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*.

ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*.

ISO 10303-21:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*.

ISO 10303-22:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 22: Implementation methods: Standard data access interface.*

ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts.*

ISO 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303-42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 10303-43:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resources: Representation structures.*

ISO 10303-44:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resources: Product structure configuration.*

ISO 10303-45:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resources: Materials.*

ISO 10303-46:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 46: Integrated generic resources: Visual presentation.*

ISO 10303-47:1997, *Industrial automation systems and integration — Product data representation and exchange — Part 47: Integrated generic resources: Shape variation tolerances.*

ISO 10303-501:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 501: Application interpreted construct: Edge-based wireframe.*

ISO 10303-502:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 502: Application interpreted construct: Shell-based wireframe.*

ISO 10303-503:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 503: Application interpreted construct: Geometrically bounded 2D wireframe.*

ISO 10303-514:1999, *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation.*

ISO 10303-515:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 515: Application interpreted construct: Constructive solid geometry.*

ISO/IEC 8824-1:1995, *Information Technology — Open Systems Interconnection — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of Basic Notation.*

IEC 60050-541:1990, *International Electrotechnical Vocabulary. Chapter 541: Printed circuits.*

IEC 61182-1:1994, *Printed boards — Electronic data description and transfer*

IEC 61346-1:1996, *Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 1: Basic rules*

ISO 129:1985, *Technical drawings — Dimensioning — General principles, definitions, methods of execution and special indications*

ISO 1101:1995, *Technical drawings — Geometrical tolerancing — Tolerancing of form, orientation, location and run-out — Generalities, definitions, symbols, indications on drawings*

ISO 2692:1988, *Technical drawings — Geometrical tolerancing — Maximum material principle*

ISO 2692-Amendment 1:1992, *Least material requirement*

ISO 8015:1985, *Technical drawings — Fundamental tolerancing principle*

ISO/IEC 9592-1:1997, *Information technology — Computer graphics and image processing — Programmer's Hierarchical Interactive Graphics System (PHIGS) — Part 1: Functional Description*

3 Terms, definitions, and abbreviations

3.1 Terms defined in IEC 60050-541

This part of ISO 10303 makes use of the following terms defined in IEC 60050-541:

- annular ring;
- blind via;
- buried via;
- cutout;
- double-sided board;
- fiducial;
- interfacial connection;
- land;
- non functional land;
- printed wiring;
- printed circuit;
- trace;
- via.

3.2 Terms defined in ISO 129

This part of ISO 10303 makes use of the following term defined in ISO 129:

- dimension.

3.3 Terms defined in ISO 1101

This part of ISO 10303 makes use of the following term defined in ISO 1101:

- tolerance zone.

3.4 Terms defined in ISO 2692

This part of ISO 10303 makes use of the following terms defined in ISO 2692:

- least material condition;
- maximum material condition;
- maximum material principle.

3.5 Terms defined in ISO 2692: Amendment 1

This part of ISO 10303 makes use of the following term defined in ISO 2692: Amendment 1:

- least material requirement.

3.6 Terms defined in ISO 5459

This part of ISO 10303 makes use of the following terms defined in ISO 5459:

- datum;
- datum system.

3.7 Terms defined in ISO 8015

This part of ISO 10303 makes use of the following terms defined in ISO 8015:

- geometric tolerance;
- tolerance.

3.8 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

- abstract test suite;
- application;
- application activity model (AAM);
- application interpreted model (AIM);
- application object;

ISO 10303-210

- application protocol (AP);
- application reference model (ARM);
- assembly;
- conformance testing;
- data;
- implementation method;
- integrated resource;
- model;
- PICS proforma;
- product;
- product data;
- unit of functionality (UoF).

3.9 Terms defined in ISO 10303-11

This part of ISO 10303 makes use of the following terms defined in ISO 10303-11:

- domain rule;
- informal proposition;
- list;
- LOGICAL;
- SELF;
- set;
- unique rule.

3.10 Terms defined in ISO 10303-44

This part of ISO 10303 makes use of the following terms defined in ISO 10303-44:

- directed acyclic graph;
- parent node.

3.11 Terms defined in ISO 10303-45

This part of ISO 10303 makes use of the following term defined in ISO 10303-45:

- data environment.

3.12 Terms defined in ISO 10303-46

This part of ISO 10303 makes use of the following terms defined in ISO 10303-46:

- font;
- tiling.

3.13 Other definitions

For the purpose of this part of ISO 10303, the following definitions apply:

3.13.1

component

an occurrence of a product used in an assembled product.

3.13.2

connected

joined; related or associated.

3.13.3

connection

the state of being connected.

3.13.4

connectivity

the state of being connected.

3.13.5

double-sided printed circuit board

a PCB with a conductive pattern on both of its sides.

3.13.6

flexible printed circuit board

a PCB that utilizes flexible insulating substrate materials with or without flexible insulating cover layers.

3.13.7

functional object

a type of product design representation described as an assembly of functions or implied function.

3.13.8

functionality

the degree to which an object is designed or adapted to a particular function or use.

3.13.9

hierarchy

a collection of related items and where the relationship between the items may be represented by a directed acyclic graph in which a node has only one parent node.

3.13.10

hierarchical decomposition

a collection of related items and where the relationship between the items may be represented by a directed acyclic graph in which a node may have one or more parent nodes.

3.13.11

insulating substrate materials

materials with a high resistance to the flow of electrical current that are used as a base for the formation of conductive patterns.

3.13.12

integrated circuit

a miniature die of substrate material that is etched or imprinted with a complex of electronic components and their interconnections.

3.13.13

interface

an access mechanism to the functionality of an object.

3.13.14

multi-layer printed circuit board

a PCB that consists of rigid or flexible insulation materials and with three or more alternate printed wiring or printed circuit layers that have been bonded together and electrically interconnected.

3.13.15

nomenclature

a list of unique string types.

3.13.16

part

a product with operational functionality that is expected to be used as a component of one or more assembled products.

EXAMPLE 1 The types of functionality are electrical, mechanical, electro-mechanical, magnetic, and optical.

3.13.17

port

an access mechanism.

3.13.18

printed circuit board layout

the geometric patterns and layered configurations of a bare printed circuit board.

3.13.19

printed component

a part manufactured during PCB fabrication.

3.13.20

rigid-flex printed circuit board

a PCB that utilizes both rigid and flexible insulating substrate materials.

3.13.21

rigid printed circuit board

a PCB that utilizes only rigid insulating substrate materials.

3.13.22

single-sided printed circuit board

a PCB with a conductive pattern on only one side.

3.13.23

structural domain

a domain in which each model explicitly states only one type of relationship between model elements.

3.14 Abbreviations

For the purposes of this part of ISO 10303, the following abbreviations apply:

2D	two-dimension
3D	three-dimension
AAM	application activity model
AIM	application interpreted model
AP	application protocol
ARM	application reference model

B-rep	boundary representation
CAD	computer-aided design
CSG	constructive solid geometry
EDA	electronic design automation
EE	electrical or electronics
EDIF	electronic design interchange format
EMI	electrical magnetic interference
IC	integrated circuit
IP	informal proposition
MCM	multi-chip module
PCA	printed circuit assembly
PCB	printed circuit board
PICS	Protocol information and conformance statement
RF	radio frequency
UoF	Units of Functionality
UR	unique rule
URL	uniform resource locator
VHDL	VHSIC hardware description language
VHSIC	very high speed integrated circuits
WR	domain rule

4 Information requirements

This clause specifies the information required for information shared between device, system, design, and manufacturing process planning engineers for transforming detailed requirements data, detailed device data, detailed manufacturing capabilities data, and detailed functional design data into analysable and manufacturable descriptions.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTE 1 A graphical representation of the information requirements is given in annex H.

NOTE 2 The information requirements correspond to those of the activities identified as being within the scope of this application protocol in annex G.

NOTE 3 The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

4.1 Units of functionality

This subclause specifies the units of functionality for the Electronic assembly, interconnect and packaging design application protocol. This Part of ISO 10303 specifies the following units of functionality:

- `advanced_boundary_representation`;
- `altered_package`;
- `altered_part`;
- `analytical_model`;
- `analytical_representation`;
- `assembly_bill_of_material`;
- `assembly_component`;
- `assembly_component_grouping`;
- `assembly_component_placement_requirements`;
- `assembly_design_information_assignment_to_usage_view`;
- `assembly_functional_interface_requirement`;

- assembly_functional_requirement_allocation;
- assembly_interface_component;
- assembly_marking;
- assembly_module_design;
- assembly_module_usage_view;
- assembly_physical_interface_requirement;
- assembly_physical_requirement_allocation;
- assembly_shape;
- assembly_technology;
- authorization;
- bare_die;
- characteristic;
- connection_zone_to_analytical_model_port_mapping;
- constructive_solid_geometry;
- datum_based_characteristic;
- datum_difference_based_characteristic;
- datum_difference_based_model;
- design_activity_control;
- design_management;
- device_marking;
- document;
- end_item_identification;
- external_definition;

- fabrication_joint;
- form_features;
- functional_decomposition_to_assembly_design;
- functional_decomposition_to_interconnect_design;
- functional_decomposition_with_nodal_representation;
- functional_decomposition_with_nodal_representation_to_assembly_mapping;
- functional_decomposition_with_nodal_representation_to_interconnect_mapping;
- functional_definition_to_physical_unit_usage_view_mapping;
- functional_definition_to_printed_part_mapping;
- functional_design_elaboration;
- functional_requirement_allocation;
- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- geometric_dimensioning_and_tolerance;
- geometrically_bounded_2d_wireframe;
- inter_stratum_feature_template;
- interconnect_bill_of_material;
- interconnect_component_grouping;
- interconnect_design_information_assignment_to_usage_view;
- interconnect_functional_requirement_allocation;
- interconnect_interface_component;

- interconnect_marking;
- interconnect_module_connection_routing;
- interconnect_module_design;
- interconnect_module_edge;
- interconnect_module_usage_view;
- interconnect_physical_requirement_allocation;
- interconnect_placement_requirements;
- interconnect_shape;
- laminate_assembly_design;
- land;
- library_item;
- manufacturing_technology;
- material_product;
- non_topological_surface;
- package;
- packaged_part_black_box_model;
- packaged_part_terminal_analytical_representation;
- packaged_part_white_box_model;
- part_feature;
- part_feature_to_analytical_model_port_mapping;
- part_template;
- part_template_shape;
- physical_network;

- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- planned_characteristic;
- planned_effectivity;
- printed_part_template;
- product_identification;
- product_requirement_allocation;
- product_rule;
- requirement_decomposition;
- sequential_laminate_assembly_design;
- shape;
- signal;
- solid_of_linear_extrusion;
- source_control;
- stratum;
- stratum_feature;
- stratum_position;
- styled_curve;
- styled_text;
- test_probe;
- wireframe_2d_model_with_topology;
- wireframe_with_topology.

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2

4.1.1 advanced_boundary_representation

The `advanced_boundary_representation` UoF defines the information needed for a representation of a product using advanced boundary representation solids models. This representation allows for the definition of curves and surfaces and the topology that bounds them. Boundaries are explicitly defined only by topology. All of the geometry that defines the product's shape shall be associated with topology.

The following application objects are used by the `advanced_boundary_representation` UoF:

- `Advanced_b_rep`.

4.1.2 altered_package

The `altered_package` UoF defines the properties of a package that has had some preparation work done to it that modifies the form or fit or material properties of the package. The preparation work is necessary to facilitate installation or support alternate mounting arrangements. The `altered_package` UoF is included to support the exchange of package related information independently of the usage of the package.

The following application objects are used by the `altered_package` UoF:

- `Altered_package`;
- `Altered_package_terminal`.

4.1.3 altered_part

The `altered_part` UoF defines the properties of a component that has had some preparation work done to it that modifies the form or fit or function or material properties of the component and is usually installed in the product to be delivered to a customer. The preparation work may be necessary to facilitate the installation, or to support alternate mounting arrangements, or to support field programmability. The `altered_part` UoF does not distinguish between the properties of a component that has been altered prior to installation and the properties of a component that has been altered, installed and delivered to a customer as part of the assembly.

NOTE Enterprises typically do not document component shape changes that are minor that occur during production of the assembly (e.g., an operation that trimmed leads would be ignored.).

The following application objects are used by the `altered_part` UoF:

- `Altered_package`;
- `Altered_package_terminal`;
- `Altered_packaged_part`;

- Engineering_make_from;
- Part_feature_make_from_relationship.

4.1.4 analytical_model

The analytical_model UoF defines the information needed to properly interface to and exchange the source code contents of an externally defined computer-interpretable model of a particular behaviour. A model allows a behaviour to be described using a mathematical form (e.g., algorithm, table). A model is independent of any particular product.

The following application objects are used by the analytical_model UoF:

- Analytical_model;
- Analytical_model_definition;
- Analytical_model_make_from_relationship;
- Analytical_model_port;
- Analytical_model_port_make_from_relationship;
- Analytical_model_scalar_port;
- Analytical_model_vector_port;
- Digital_analytical_model_port;
- Language_reference_manual;
- Model_parameter;
- Terminal_pair_port_equivalence_relationship.

4.1.5 analytical_representation

The analytical_representation UoF defines the information needed to associate specific instance data with an analytical model to support computer-based experimentation. The models are associated with a network representation in order to form a complete system of equations for numerical or symbolic analysis. Each discipline may have a different analytical representation of a product design.

NOTE The network representation commonly used, a nodal representation is supported by the functional_decomposition_with_nodal_representation UoF.

The following application objects are used by the analytical_representation UoF:

- Analytical_representation.

4.1.6 assembly_bill_of_material

The assembly_bill_of_material UoF defines the information needed to describe the structured list of materials of components required for an assembly. The assembly_bill_of_material UoF includes the vendor supplying the product, all approved alternates and substitutes, and which product selection is dependent on the result of a test of the assembly.

The following application objects are used by the assembly_bill_of_material UoF:

- Alternate_product;
- Alternate_select_product;
- Assembly_alternate_product;
- Assembly_composition_relationship;
- Assembly_material_composition_relationship;
- Next_higher_assembly_relationship;
- Physical_component;
- Promissory_usage_in_product_model_relationship;
- Promissory_usage_relationship;
- Substitute_product;
- Test_select_product.

4.1.7 assembly_component

The assembly_component UoF defines the information needed to incorporate occurrences of the different types of physical units as components in an assembly. The assembly_component UoF defines the information needed to incorporate components that use the different types of part templates as definitions in an assembly. The assembly_component UoF defines auxiliary roles a component may assume in an assembly in support of other components in addition to the primary role of component. The assembly_component UoF functionality supports discrete and fabricated assemblies.

The following application objects are used by the `assembly_component` UoF:

- `Assembly_component`;
- `Assembly_module_component`;
- `Assembly_module_macro_component`;
- `Assembly_module_macro_component_join_terminal`;
- `Bare_die_component`;
- `Cable_component`;
- `Interconnect_module_component`;
- `Interconnect_module_macro_component`;
- `Interface_component`;
- `Packaged_component`;
- `Packaged_connector_component`;
- `Physical_component`;
- `Routed_interconnect_component`;
- `Shield`;
- `Thermal_component`.

4.1.8 `assembly_component_grouping`

The `assembly_component_grouping` UoF defines the information needed to describe the different ways of grouping assembly components in an assembly. The `assembly_component_grouping` UoF functionality supports discrete and fabricated assemblies. The `assembly_component_grouping` UoF functionality supports design reuse of complex design elements.

NOTE The `assembly_component_grouping` UoF is not intended to be the means to support hierarchical design, as that is the purpose of the `assembly_module_design` and `interconnect_module_design` UoFs.

The following application objects are used by the assembly_component_grouping UoF:

- Array_placement_group;
- Assembly_component_placement_link;
- Assembly_group_component;
- Interfaced_group_component;
- Linear_array_placement_group_component;
- Linear_array_placement_group_component_link;
- Physical_component;
- Rectangular_array_placement_group_component;
- Sub_assembly_relationship.

4.1.9 assembly_component_placement_requirements

The assembly_component_placement_requirements UoF defines the information needed to describe the constraints restricting where components are to be placed in an assembly. These restrictions may be component category to board feature category or component category to component category related. Application object attribute data population is restricted to that related to the discrete assembly view.

The following application objects are used by the assembly_component_placement_requirements UoF:

- Component_group_assignment;
- Component_placement_restriction_assignment;
- Interconnect_module_component_surface_feature;
- Item_restricted_requirement_occurrence;
- Layout_spacing_requirement_occurrence;
- Mounting_restriction_area;
- Mounting_restriction_volume;
- Placement_group;
- Placement_group_area_assignment;

- Placement_group_volume_assignment.

4.1.10 assembly_design_information_assignment_to_usage_view

The assembly_design_information_assignment_to_usage_view UoF defines the information that describes the association between product data necessary for a design view and product data necessary to define a usage view. A product engineer may choose to only release usage view information in order to allow other designers to incorporate the product into a higher level assembly. This information is developed and maintained by a design organization in order to control the accuracy and integrity of externally released data.

The following application objects are used by the assembly_design_information_assignment_to_usage_view UoF:

- Component_feature_to_physical_usage_view_assignment;
- Component_terminal_to_assembly_module_interface_terminal_assignment;
- Component_terminal_to_assembly_module_join_terminal_assignment;
- Component_terminal_to_pca_terminal_assignment.

4.1.11 assembly_functional_interface_requirement

The assembly_functional_interface_requirement UoF defines the information needed to describe the manner by which an electronic assembly shall electrically or functionally interface to the next higher assembly. Constraints on the signals associated with terminals of the mating connector may be specified. Identification of the mating terminals is supported.

The following application objects are used by the assembly_functional_interface_requirement UoF:

- Mating_connector_termination;
- Termination_constraint.

4.1.12 assembly_functional_requirement_allocation

The assembly_functional_requirement_allocation UoF describes the assignment of network based functional requirements to an assembly under design. These requirements are specific to the relationship between the functional connectivity definition and the component instances used in the assembly under design that implement that connectivity.

NOTE Refer to the functional_requirement_allocation UoF for information assigning textual requirements to a functional decomposition.

The following application objects are used by the `assembly_functional_requirement_allocation` UoF:

- `Physical_component_terminal`;
- `Physical_node_requirement_to_implementing_component_allocation`;
- `Test_requirement_allocation`.

4.1.13 `assembly_interface_component`

The `assembly_interface_component` UoF defines the information needed to describe the component that acts as a connector that provides external access to an assembly. The items' position in the assembly, and the specification that dictates the selection and placement are included.

The following application objects are used by the `assembly_interface_component` UoF:

- `Ee_specification`;
- `Interface_component`;
- `Packaged_connector_component`.

4.1.14 `assembly_marking`

The `assembly_marking` UoF defines the information needed to apply identification information, using material, to an assembly. The content of the information is under configuration control. The scope of this UoF is for assemblies.

The following application objects are used by the `assembly_marking` UoF:

- `Configuration_controlled_message`;
- `Ee_specification`;
- `Marking`.

4.1.15 `assembly_module_design`

The `assembly_module_design` UoF defines the information needed to describe the components in an assembly and the physical relationship between the components and the assembly. The `assembly_module_design` UoF defines the information needed to describe the different features of components and the relationship between component features in an assembly. The `assembly_module_design` UoF defines the information needed to describe the materials used in an assembly that physically support components. Analysis data extraction is supported to allow verification of physical connectivity implementation.

The following application objects are used by the assembly_module_design UoF:

- Assembled_with_bonding;
- Assembled_with_fasteners;
- Assembly_component;
- Assembly_composition_relationship;
- Assembly_joint;
- Assembly_material_composition_relationship;
- Assembly_module;
- Assembly_module_component_terminal;
- Assembly_module_macro_component;
- Assembly_module_macro_component_join_terminal;
- Bare_die_component_terminal;
- Component_feature;
- Component_feature_make_from_relationship;
- Component_feature_relationship;
- Component_interface_terminal;
- Component_make_from_relationship;
- Component_mating_constraint_condition;
- Component_mounting_clearance_relationship;
- Component_mounting_feature;
- Component_overlap_relationship;
- Component_terminal;
- Connection_zone_based_assembly_joint;

- Interconnect_component_interface_terminal;
- Interconnect_component_join_terminal;
- Interconnect_module_component_surface_feature;
- Interconnect_module_component_terminal;
- Interconnect_module_macro_component;
- Interface_mounted_join;
- Movable_packaged_component_join_terminal;
- Next_higher_assembly_relationship;
- Package_terminal_occurrence;
- Packaged_component_join_terminal;
- Packaged_connector_component_interface_terminal;
- Pca;
- Physical_component;
- Routed_physical_component.

4.1.16 assembly_module_usage_view

The assembly_module_usage_view UoF defines the information needed to describe the electronic assembly for use as a component at the next level of assembly. The level of information provided is for use of the electronic assembly and not manufacture.

The following application objects are used by the assembly_module_usage_view UoF:

- Assembly_module_interface_terminal;
- Assembly_module_join_terminal;
- Assembly_module_terminal;
- Assembly_module_usage_view;
- Assembly_module_usage_view_connector_relationship;

- Packaged_connector;
- Pca_terminal;
- Pca_usage_view.

4.1.17 assembly_physical_interface_requirement

The assembly_physical_interface_requirement UoF defines the information needed to describe the manner by which an electronic assembly shall mechanically interface to the next higher assembly. The type of mating connector and its location relative to the electronic assembly are provided as a design requirement, that must be satisfied in the design.

The following application objects are used by the assembly_physical_interface_requirement UoF:

- Component_placement_restriction_assignment;
- Interface_requirement_occurrence;
- Mating_connector_component;
- Mating_connector_placement_relationship;
- Next_higher_assembly_interface_requirement.

4.1.18 assembly_physical_requirement_allocation

The assembly_physical_requirement_allocation UoF describes the assignment of customer, regulatory agency, and enterprise requirements to an assembly under design. These requirements may be assigned to physical items or geometric features that are used in various phases of the design process.

The following application objects are used by the assembly_physical_requirement_allocation UoF:

- Assembly_requirement_allocation;
- Assembly_separation_requirement_allocation;
- Electromagnetic_compatibility_requirement_allocation;
- Interface_component_allocation;
- Requirement_allocation;
- Shield_allocation;
- Thermal_requirement_allocation.

4.1.19 assembly_shape

The assembly_shape UoF defines the information needed to describe explicit geometric models and transformations used in the design process for an electronic assembly. This UoF permits a different type of geometric representation for a component in a library and for an instance of the component in the design.

The following application objects are used by the assembly_shape UoF:

- Assembly_component_2d_shape;
- Assembly_component_3d_shape;
- Bond_assembly_position;
- Component_2d_edge_location;
- Component_2d_location;
- Component_2d_stacked_location;
- Component_2d_surface_location;
- Component_3d_location;
- Component_assembly_2d_position;
- Component_assembly_3d_position;
- Component_part_2d_geometric_representation_relationship;
- Component_part_3d_geometric_representation_relationship;
- Design_view_connection_zone_position_relationship;
- Design_view_terminal_component_shape_relationship.

4.1.20 assembly_technology

The assembly_technology UoF defines the information needed to describe the different aspects of how components are joined together to make an assembly. The assembly_technology UoF defines the information needed to describe the material used in making the physical connections and their typical shape.

The following application objects are used by the assembly_technology UoF:

- Assembly_bond_definition;
- Assembly_technology_specification;
- Default_assembly_bond_shape;
- Reference_packaged_part_assembly_implementation;
- Reference_packaged_part_interconnect_implementation.

4.1.21 authorization

The authorization UoF defines the information necessary to determine the acceptance of a piece of product data within an enterprise, on a specific date, at a particular level of status, by a group of people having roles that allow them to grant the authorization.

The following application objects are used by the authorization UoF:

- Contract;
- Date;
- Date_and_time;
- Ee_approval;
- Organization;
- Person;
- Person_and_organization;
- Security_classification.

4.1.22 bare_die

The bare_die UoF defines the information needed to describe a semiconductor material product that may reside inside an integrated circuit, or that may be a discrete active component, or that may be included as a component in an electronic assembly. Terminal information is supported, including explicit shape data.

The following application objects are used by the bare_die UoF:

- Bare_die;
- Bare_die_bottom_surface;
- Bare_die_edge_segment_surface;
- Bare_die_edge_surface;
- Bare_die_surface;
- Bare_die_terminal;
- Bare_die_terminal_surface_constituent_relationship;
- Bare_die_top_surface;
- Connection_zone_interface_plane_relationship;
- Guided_wave_bare_die_terminal;
- Minimally_defined_bare_die_terminal;
- Seating_plane.

4.1.23 characteristic

The characteristic UoF defines the information needed to describe parameters and tolerances for a physical unit, a template, a design, or a model. Explicit units, as well as exact and uncertain measures may be described.

The following application objects are used by the characteristic UoF:

- Absorbed_dose_data_element;
- Activity_data_element;
- Amount_of_substance_data_element;
- Angle_data_element;
- Area_data_element;
- Capacitance_data_element;

- Celsius_temperature_data_element;
- Characteristic;
- Characteristic_data_set;
- Characteristic_data_table;
- Compound_characteristic;
- Conductance_data_element;
- Coordinated_characteristic;
- Data_element;
- Dose_equivalent_data_element;
- Ee_colour;
- Electric_charge_data_element;
- Electric_current_data_element;
- Electromotive_force_data_element;
- Energy_data_element;
- Event_data_element;
- Expanded_uncertainty;
- Force_data_element;
- Frequency_data_element;
- Illuminance_data_element;
- Inductance_data_element;
- Length_data_element;
- Luminous_flux_data_element;
- Luminous_intensity_data_element;

- Magnetic_flux_data_element;
- Magnetic_flux_density_data_element;
- Mass_data_element;
- Maximum_tolerance_characteristic;
- Measure_qualification;
- Minimum_tolerance_characteristic;
- Model_parameter;
- Numeric_parameter;
- Parameter_assignment;
- Pitch_class;
- Plus_minus_tolerance_characteristic;
- Power_data_element;
- Precision_qualifier;
- Pressure_data_element;
- Qualitative_uncertainty;
- Range_characteristic;
- Ratio_data_element;
- Resistance_data_element;
- Scattering_parameter_data_element;
- Solid_angle_data_element;
- Standard_uncertainty;
- Statistical_tolerance_characteristic;
- Symmetrical_tolerance_characteristic;

- Thermodynamic_temperature_data_element;
- Time_data_element;
- Tolerance_characteristic;
- Type_qualifier;
- Uncertainty_qualifier;
- Volume_data_element.

4.1.24 connection_zone_to_analytical_model_port_mapping

The connection_zone_to_analytical_model_port_mapping UoF defines the information needed to associate specific geometric founded data for a physical unit to an analytical model port to support computer-based experimentation on a specific product design. Each discipline may have a different connection zone to analytical model port mapping for a product design. Direct specification of the mapping to a connection area on the terminal of the product concepts under investigation is provided. This UoF provides additional level of detail over that found in the part_feature_to_analytical_model_port_mapping UoF.

The following application objects are used by the connection_zone_to_analytical_model_port_mapping UoF:

- Connection_zone_model_port_assignment.

4.1.25 constructive_solid_geometry

The constructive_solid_geometry UoF defines the information needed to describe the solid representation of an item by use of primitive solid construction elements. These primitive elements may be intersected, unioned, and differenced with each other in order, to form desired representation. The construction history of the solid is included by default.

The following application objects are used by the constructive_solid_geometry UoF:

- Block;
- Boolean_csg_result;
- Csg_primitive;
- Csg_solid;
- Right_angular_wedge;
- Right_circular_cylinder.

4.1.26 datum_based_characteristic

The datum_based_characteristic UoF defines the information needed to describe parameters for a physical unit, a template, a design, or a model, where the characteristic is based on a Datum.

The following application objects are used by the datum_based_characteristic UoF:

- Datum_based_length_measure.

4.1.27 datum_difference_based_characteristic

The datum_difference_based_characteristic UoF defines the information needed to describe parameters for a physical unit, a template, a design, or a model where the characteristic is based on a Datum_difference.

The following application objects are used by the datum_difference_based_characteristic UoF:

- Datum_difference;
- Datum_difference_based_characteristic.

4.1.28 datum_difference_based_model

The datum_difference_based_model UoF defines the information needed to properly interface to and exchange an externally defined computer-interpretable model of a particular product that includes interfaces based on Datums. A datum_difference_based_model allows a product's characteristics and performance parameters to be described analytically and referenced to a base or reference computer interpretable geometric model. A datum_difference_based_model is an extension of the Analytical_model UoF.

The following application objects are used by the datum_difference_based_model UoF:

- Analytical_model;
- Analytical_model_port;
- Analytical_model_scalar_port;
- Analytical_model_vector_port;
- Datum_difference;
- Datum_difference_analytical_model_port_assignment;
- Datum_difference_based_model_parameter;

- Datum_difference_functional_unit_usage_view_terminal_assignment;
- Digital_analytical_model_port;
- Language_reference_manual;
- Model_parameter;
- Terminal_pair_port_equivalence_relationship.

4.1.29 design_activity_control

The design_activity_control UoF defines the information necessary to describe the history of product versions. It identifies initial product design requests, as well as change requests for revised product versions. It provides the directive for work to proceed in the development of these initial or modified product versions.

The following application objects are used by the design_activity_control UoF:

- Change_order;
- Change_request;
- Start_order;
- Start_request;
- Work_order;
- Work_request.

4.1.30 design_management

The design_management UoF defines the information needed to describe the association between an original product definition and a planned product definition. The planned product definition is based upon the original definition.

The following application objects are used by the design_management UoF:

- Add_design_object_management_relationship;
- Change_design_object_management_relationship;
- Delete_design_object_management_relationship;

- Design_object_management_relationship;
- Frozen_design_object;
- Managed_design_object;
- Owned_design_object;
- Owner_assignment.

4.1.31 device_marking

The device_marking UoF defines the information needed to apply identification information, using material, to an item used in a design for identification of that item. The content of the information is under configuration control. The scope of this UoF is for library data.

The following application objects are used by the device_marking UoF:

- Configuration_controlled_message;
- Ee_specification;
- Marking.

4.1.32 document

The document UoF defines the information needed to describe a set of different types of information elements that are found in a document that are used to specify, in detail, a product or an aspect of a product or model. Support is provided for describing text that is a part of a document.

EXAMPLE 1 Source code for models or requirements is data which may be exchanged using the capabilities of this UoF.

The following application objects are used by the document UoF:

- Design_specification;
- Ee_document;
- Ee_document_identifier;
- Ee_specification;
- Ee_text;
- Fabrication_technology_specification;

- Interface_specification;
- Language_reference_manual;
- Lead_form_specification;
- Material_specification;
- Process_specification;
- Structured_text;
- Surface_finish_specification;
- Test_specification.

4.1.33 end_item_identification

The end_item_identification UoF defines the information needed to describe the version of the product that is being provided to satisfy the customer's requirements.

The following application objects are used by the end_item_identification UoF:

- Ee_product_configuration;
- Ee_product_model.

4.1.34 external_definition

The external_definition UoF defines the information needed to describe any definitions that are not predefined in this part of ISO 10303. The external definition requires a computer interpretable data dictionary for the most effective implementation.

The following application objects are used by the external_definition UoF:

- External_definition.

4.1.35 fabrication_joint

The fabrication_joint UoF defines the information needed to describe a joint created during the interconnect substrate manufacturing process. The joint purpose is to describe an intended connection between the interconnect substrate materials. The joint concept may be for strictly design or analysis purposes and may not be resolvable as a separate feature in the finished product.

The following application objects are used by the fabrication_joint UoF:

- Connection_zone_based_fabrication_joint;
- Fabrication_joint;
- Interconnect_module_product_assembly_view_fabrication_joint.

4.1.36 form_features

The form_features UoF defines the information needed to describe certain explicitly modeled parametrically defined geometric feature definitions provided by this part of ISO 10303.

The following application objects are used by the form_features UoF:

- Bevel_edge_feature_shape;
- Blind_passage_template;
- Counterbore_passage_template;
- Countersunk_passage_template;
- Datum;
- Edge_segment_cross_section;
- Edge_shape;
- Interconnect_module_edge_segment_surface;
- Interconnect_module_primary_surface;
- Interconnect_module_secondary_surface;
- Interconnect_module_surface_feature;
- Radius_edge_feature_shape;
- Step_edge_feature_shape.

4.1.37 functional_decomposition_to_assembly_design

The functional_decomposition_to_assembly_design UoF defines the information needed to describe the assignment of a functional unit in a network definition to an element in the assembly. Usually, this information is provided by a packaging application. This UoF defines the different means of describing swapping accomplished within a design. The types of swapping covered are the ones between gates in a single package instance, between gates in different package instances, and between terminals in a gate.

NOTE Constraints on swapping are not included in this UoF. The terminals swapped with this UoF are on the functional elements in the hierarchical design network listing and not the flattened network listing. Applications that perform swapping on flattened representations will need to map to the hierarchical representation.

The following application objects are used by the functional_decomposition_to_assembly_design UoF:

- Component_swap_specification;
- Design_composition_path;
- Design_functional_unit_allocation_to_assembly_component;
- Design_functional_unit_allocation_to_assembly_joint;
- Gate_path_swap_specification;
- Terminal_swap_specification.

4.1.38 functional_decomposition_to_interconnect_design

The functional_decomposition_to_interconnect_design UoF defines the information needed to describe the assignment of a functional unit in a network definition to an element in the Interconnect_module. The functional_decomposition_to_interconnect_design UoF defines the different means of performing swapping within a design. The types of swapping covered are the ones between gates in a single printed part, between identical gates in different instances of a printed part, and between terminals of identical type and behaviour in a gate.

The following application objects are used by the functional_decomposition_to_interconnect_design UoF:

- Component_swap_specification;
- Design_composition_path;
- Design_functional_unit_allocation_to_assembly_component;
- Design_functional_unit_allocation_to_fabrication_joint;

- Gate_path_swap_specification;
- Physical_component_terminal;
- Terminal_swap_specification;
- Transmission_line_functional_unit_terminal_allocation;
- Transmission_line_functional_unit_to_routed_printed_component_allocation.

4.1.39 functional_decomposition_with_nodal_representation

The functional_decomposition_with_nodal_representation UoF defines the information needed to describe how a set of functional units are connected together to form a network. The individual connections may either be discrete or may be part of a defined bus structure. The network may be specified as electrical, thermal, or unspecified (to be resolved by the simulation environment).

NOTE The analytical_representation and analytical_model UoFs are also needed to build a complete set of input data for a simulation experiment, as this UoF only provides the network structure.

The following application objects are used by the functional_decomposition_with_nodal_representation UoF:

- Bus_element_link;
- Bus_structural_definition;
- Electrical_network_definition;
- Functional_unit;
- Functional_unit_definition;
- Functional_unit_network_definition;
- Functional_unit_network_node_definition;
- Functional_unit_network_terminal_definition_bus_assignment;
- Functional_unit_network_terminal_definition_node_assignment;
- Functional_unit_terminal;
- Functional_unit_terminal_bus_assignment;
- Functional_unit_terminal_node_assignment;

- Functional_unit_usage_view;
- Functional_unit_usage_view_terminal_definition;
- Thermal_network_definition.

4.1.40 functional_decomposition_with_nodal_representation_to_assembly_mapping

The functional_decomposition_with_nodal_representation_to_assembly_mapping UoF defines the information necessary to describe the association between the multi-level hierarchical nodal representation and the single level nodal representation described by a physical_connectivity_definition. Multi-level hierarchy is represented in this part of ISO 10303 by a graph, and there are several graph instances involved in this UoF. The lowest level primitives in the design graph are also the lowest level primitives in the functional definition of a physical unit. A mapping is defined between paths contained in two graph instances in order to define equivalent functionality for a design element and a library element that contributes to the functionality of a physical_unit or part_template whose definition is in a library. Information is provided to exchange the constraints determined on the allowed alternatives that may be selected by a mapping activity. The constraints may be applied to individual elements and also to individual terminals.

EXAMPLE 1 A packaging application that used this mapping would assign element names "A,B,C" of a multi-element part to three path instances in the design. A second mapping is defined to allow the packaging application to associate the reference designator of the assembly_component to the three paths through the design graph that are implemented with that assembly_component. It may be determined by the design activity that element "A" is permitted to be exchanged with element "B" but not with element "C".

The following application objects are used by the functional_decomposition_with_nodal_representation_to_assembly_mapping UoF:

- Aggregate_connectivity_requirement;
- Connectivity_allocation;
- Design_composition_path;
- Design_functional_unit_allocation_to_assembly_component;
- Design_functional_unit_allocation_to_reference_functional_unit;
- Functional_usage_view_to_part_terminal_assignment;
- Minimally_defined_component_terminal;
- Package_external_reference;
- Package_terminal_external_reference;

- Part_terminal_external_reference;
- Physical_component_terminal;
- Reference_composition_path;
- Reference_functional_unit_assignment_to_part.

4.1.41 functional_decomposition_with_nodal_representation_to_interconnect_mapping

The functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF defines the information necessary to describe the association between the multi-level hierarchical nodal representation and the single level nodal representation described by a physical_connectivity_definition that is in the design view of an interconnect. Multi-level hierarchy is represented in this part of ISO 10303 by a graph, and there are several graph instances involved in this UoF. The lowest level primitives in the design graph are also the lowest level primitives in the functional definition of a part template. A mapping is defined between paths contained in two graph instances in order to define equivalent functionality for a design element and a library element that contributes to the functionality of a part_template whose definition is in a library. Information is provided to exchange the constraints determined on the allowed alternatives that may be selected by a mapping activity. The constraints may be applied to individual elements and also to individual terminals.

EXAMPLE 1 A packaging application that used this mapping would assign element names "A,B,C" of a multi-element part to three path instances in the design. A second mapping is defined to allow the packaging application to associate the reference designator of the assembly_component to the three paths through the design graph that are implemented with that assembly_component. It may be determined by the design activity that element "A" is permitted to be exchanged with element "B" but not with element "C".

NOTE The population of assembly_component supported by this UoF is the types of laminate_component.

The following application objects are used by the functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF:

- Aggregate_connectivity_requirement;
- Connectivity_allocation;
- Design_composition_path;
- Design_functional_unit_allocation_to_assembly_component;
- Design_functional_unit_allocation_to_reference_functional_unit;
- Functional_usage_view_to_part_terminal_assignment;
- Physical_component_terminal;

- Reference_composition_path;
- Reference_functional_unit_assignment_to_part.

4.1.42 functional_definition_to_physical_unit_usage_view_mapping

The functional_definition_to_physical_unit_usage_view_mapping UoF defines the association between functional elements and the physical units that exhibit the behaviour defined in the functional element. The functional_definition_to_physical_unit_usage_view_mapping UoF defines minimal information needed for classifying the association to support design alternative selection (e.g., swapping) during the detailed design activity.

The following application objects are used by the functional_definition_to_physical_unit_usage_view_mapping UoF:

- Functional_usage_view_to_part_terminal_assignment;
- Reference_composition_path;
- Reference_functional_unit_assignment_to_part.

4.1.43 functional_definition_to_printed_part_mapping

The functional_definition_to_printed_part_mapping UoF defines the association between a functional element and a printed part template that exhibits the behaviour defined in the functional definition. The functional_definition_to_printed_part_mapping UoF defines minimal information needed for classifying the association to support design alternative selection (e.g., swapping) during the detailed design activity.

The following application objects are used by the functional_definition_to_printed_part_mapping UoF:

- Functional_usage_view_to_printed_part_template_terminal_assignment;
- Reference_composition_path;
- Reference_functional_unit_assignment_to_part.

4.1.44 functional_design_elaboration

The functional_design_elaboration UoF defines the information needed to describe the structure of a network realization (flattened listing). It furthermore states the requirements which dictated the topological design of the realized network. It states constraints on exchange of terminal assignments to achieve the desired functionality.

EXAMPLE 1 A constraint may be a swapping constraint to ensure required fan-in capability.

The following application objects are used by the functional_design_elaboration UoF:

- Minimally_defined_component_terminal;
- Part_terminal_external_reference;
- Physical_component_terminal;
- Physical_connectivity_definition;
- Physical_connectivity_definition_relationship;
- Physical_connectivity_element;
- Physical_connectivity_make_from_relationship;
- Physical_connectivity_structure_definition;
- Physical_unit_network_definition;
- Topological_junction.

4.1.45 functional_requirement_allocation

The functional_requirement_allocation UoF describes the assignment of customer, regulatory agency, and enterprise requirements to a functional element in a network under design or to a functional specification. These requirements may be assigned for various purposes and phases of the design process.

The following application objects are used by the functional_requirement_allocation UoF:

- Functional_unit_requirement_allocation.

4.1.46 functional_specification

The functional_specification UoF defines the information needed to describe the measured behaviour of a functional element under a set of controlled conditions. It also defines the information needed to describe equivalents among different functional elements.

The following application objects are used by the functional_specification UoF:

- Equivalent_functional_unit_definition_assignment;
- Functional_specification;
- Functional_specification_definition;
- Signal_specification.

4.1.47 functional_unit_analytical_representation

The functional_unit_analytical_representation UoF defines the information needed to associate specific functional definitions with an analytical model to support computer-based experimentation. Each analysis model port is explicitly mapped to a terminal of the functional definition. Domain specific analysis is supported by a predefined type system and predefined units for simulation variables.

The following application objects are used by the functional_unit_analytical_representation UoF:

- Analytical_model_port_assignment;
- Analytical_representation.

4.1.48 functional_usage_view

The functional_usage_view UoF defines the information needed to describe the characterized behaviour and interface of a functional element. The terminals of the functional element may be assigned type information to facilitate a swapping application.

The following application objects are used by the functional_usage_view UoF:

- Equivalent_functional_terminals_assignment;
- Functional_terminal_group;
- Functional_terminal_group_assignment;
- Functional_unit_usage_view;
- Functional_unit_usage_view_terminal_definition;
- Functional_unit_usage_view_terminal_definition_make_from_relationship;
- Scalar_terminal_definition;
- Scalar_terminal_definition_link.

4.1.49 functional_usage_view_terminal_to_analytical_model_port_mapping

The functional_usage_view_terminal_to_analytical_model_port_mapping UoF defines the information needed to describe the assignment between a terminal of a functional element and a port of an analytical model. This is used in analysing a design. This UoF supports assigning explicit behaviour that is defined in a simulation model to a functional element.

NOTE The behavioural assignment is a consequence of the assignment of the terminals between the functional element and the simulation model.

The following application objects are used by the functional_usage_view_terminal_to_analytical_model_port_mapping UoF:

- Analytical_model_port_assignment.

4.1.50 geometric_dimensioning_and_tolerance

The geometric_dimensioning_and_tolerance UoF defines the information required to describe a behavioural model of a physical_unit and associated properties using geometric dimensioning and tolerancing principles in accordance with pre-existing standards.

The following application objects are used by the geometric_dimensioning_and_tolerance UoF:

- Angular_dimension_with_direction_vector;
- Angular_size_dimension;
- Angularity_tolerance;
- Apex;
- Axis_placement;
- Axis_placement_2d;
- Axis_placement_3d;
- Boundary_based_position_tolerance;
- Boundary_size_characteristic;
- Boundary_zone_definition_with_specified_size;
- Centre_axis;
- Centre_of_symmetry;
- Centre_plane;
- Centre_point;
- Circular_runout_tolerance;
- Circularity_tolerance;
- Common_datum;

- Composite_feature_locating_position_tolerance;
- Composite_feature_relating_position_tolerance;
- Composite_group_shape_element;
- Composite_linear_profile_locating_tolerance;
- Composite_linear_profile_refinement_tolerance;
- Composite_shape_element;
- Composite_surface_profile_locating_tolerance;
- Composite_surface_profile_refinement_tolerance;
- Composite_unit_shape_element;
- Concentricity_and_coaxiality_tolerance;
- Conical_tolerance_zone_boundary;
- Conical_tolerance_zone_boundary_and_surface_relationship;
- Conical_tolerance_zone_boundary_definition;
- Conical_tolerance_zone_boundary_diametrical_size_characteristic;
- Conical_tolerance_zone_definition;
- Curve_dimension;
- Cylindricity_tolerance;
- Datum;
- Datum_axis;
- Datum_axis_and_tolerance_zone_orienting_relationship;
- Datum_axis_related_orientation;
- Datum_based_direction_vector;
- Datum_based_vector_orientation;

- Datum_feature;
- Datum_feature_material_condition_property;
- Datum_feature_usage_in_common_datum_relationship;
- Datum_feature_usage_in_datum_relationship;
- Datum_feature_usage_in_datum_system_relationship;
- Datum_feature_usage_in_single_datum_relationship;
- Datum_plane;
- Datum_point;
- Datum_precedence_assignment;
- Datum_reference_frame;
- Datum_system;
- Datum_system_defining_conditions;
- Datum_system_defining_conditions_with_material_conditions;
- Datum_system_defining_conditions_without_material_conditions;
- Datum_target;
- Datum_target_set;
- Datum_target_usage_in_datum_target_set_relationship;
- Datum_usage_in_datum_system_relationship;
- Derived_shape_element;
- Diametrical_boundary_size_characteristic;
- Dimension_along_curve;
- Dimension_location_with_direction_vector;
- Dimension_location_with_specified_datum_system;

- Dimension_qualifier;
- Dimension_related_positional_boundary;
- Dimension_related_positional_boundary_definition;
- Dimension_set;
- Dimensional_characteristic;
- Dimensional_characteristic_representation;
- Dimensional_location;
- Dimensional_size;
- Dimensional_size_based_on_opposing_boundaries;
- Directed_axis;
- Direction_element;
- Externally_defined_dimension_representation;
- Flatness_tolerance;
- Geometric_alignment;
- Geometric_intersection;
- Geometric_tolerance;
- Geometric_tolerance_qualifier;
- Geometric_tolerance_with_applied_material_condition_property;
- Geometric_tolerance_with_referenced_datum_system;
- Geometric_tolerance_without_referenced_datum_system;
- Inter_stratum_feature;
- Limit_dimension_representation;
- Linear_extension;

- Maximum_dimension_representation;
- Minimum_dimension_representation;
- Non_feature_shape_element;
- Non_feature_shape_element_relationship;
- Opposing_boundary_set_size_characteristic;
- Parallel_shape_element_relationship;
- Parallelism_tolerance;
- Part_feature;
- Perpendicular_shape_element_relationship;
- Perpendicularity_tolerance;
- Physical_component_feature;
- Physical_feature_or_part_template;
- Physical_laminate_component;
- Plus_minus_tolerance_characteristic;
- Plus_minus_tolerance_range_representation;
- Position_tolerance;
- Position_tolerance_with_referenced_datum_system;
- Position_tolerance_without_referenced_datum_system;
- Positional_boundary;
- Positional_boundary_and_profile_boundary_member_relationship;
- Positional_boundary_definition;
- Positional_boundary_member;
- Positional_boundary_member_definition;

- Positional_boundary_offset;
- Pre_defined_dimension_qualifier;
- Pre_defined_geometric_tolerance_qualifier;
- Profile_boundary_definition;
- Profile_boundary_definition_with_offsets;
- Profile_of_any_line_tolerance;
- Profile_of_any_line_tolerance_with_referenced_datum_system;
- Profile_of_any_line_tolerance_without_referenced_datum_system;
- Profile_of_any_surface_tolerance;
- Profile_of_any_surface_tolerance_with_referenced_datum_system;
- Profile_of_any_surface_tolerance_without_referenced_datum_system;
- Profile_related_positional_boundary;
- Profile_related_positional_boundary_definition;
- Projected_tolerance_zone_definition;
- Projected_zone_and_base_relationship;
- Projected_zone_height_characteristic;
- Restraint_condition;
- Runout_tolerance_zone_definition_with_specified_angle;
- Runout_tolerance_zone_orienting_relationship;
- Runout_zone_orientation;
- Separate_geometric_tolerance_requirement;
- Shape_dimension_representation;
- Shape_element;

- Shape_element_boundary_relationship;
- Shape_element_composing_relationship;
- Shape_element_constituent_relationship;
- Shape_element_deriving_relationship;
- Shape_element_locating_relationship;
- Shape_element_relationship;
- Simultaneous_geometric_tolerance_requirement;
- Single_boundary_dimension;
- Single_datum;
- Singular_dimension_representation;
- Statistical_dimensional_tolerance;
- Statistical_geometric_tolerance;
- Straightness_tolerance;
- Stratum_feature;
- Symmetry_tolerance;
- Tangent_plane;
- Tangent_shape_element;
- Theoretically_exact_dimension_representation;
- Tolerance_specific_restraint_condition;
- Tolerance_zone;
- Tolerance_zone_and_viewing_plane_orienting_relationship;
- Tolerance_zone_boundary;
- Tolerance_zone_boundary_definition;

- Tolerance_zone_boundary_member;
- Tolerance_zone_boundary_member_offset;
- Tolerance_zone_circular_or_cylindrical_or_spherical_boundary;
- Tolerance_zone_definition;
- Tolerance_zone_definition_with_per_unit_size_specification;
- Tolerance_zone_definition_with_specified_orientation;
- Tolerance_zone_explicit_opposing_boundary_set;
- Tolerance_zone_form;
- Tolerance_zone_implicit_opposing_boundary_set;
- Tolerance_zone_opposing_boundary_set;
- Tolerance_zone_per_unit_size_specification;
- Total_runout_tolerance;
- True_profile;
- User_defined_dimension_qualifier;
- User_defined_geometric_tolerance_qualifier;
- Viewing_plane;
- Viewing_plane_based_orientation;
- Viewing_plane_definition.

4.1.51 geometrically_bounded_2d_wireframe

The `geometrically_bounded_2d_wireframe` UoF defines the information needed to describe the representation of the shape of a product by 2 dimensional wireframes that are defined by geometry. This includes 2D curves.

The following application objects are used by the `geometrically_bounded_2d_wireframe` UoF:

- `geometrically_bounded_2d_wireframe`.

4.1.52 inter_stratum_feature_template

The `inter_stratum_feature_template` UoF defines the information needed to describe product data, including shape data and terminal definitions. Occurrences of this data are used to describe electrical connectivity, and other design relationships, between different layers of an interconnect substrate. This UoF is used in conjunction with `manufacturing_technology` UoF.

The following application objects are used by the `inter_stratum_feature_template` UoF:

- `Component_termination_passage_template`;
- `Component_termination_passage_template_interface_terminal`;
- `Component_termination_passage_template_join_terminal`;
- `Component_termination_passage_template_terminal`;
- `Counterbore_passage_template`;
- `Countersunk_passage_template`;
- `Inter_stratum_feature_template`;
- `Unsupported_passage_template`;
- `Via_template`;
- `Via_template_terminal`.

4.1.53 interconnect_bill_of_material

The `interconnect_bill_of_material` UoF defines the information needed to describe the structured list of materials of components required to make an interconnect. The `interconnect_bill_of_material` UoF includes the vendor supplying the product, all approved alternates and substitutes, and which product selection is dependent on the result of a test.

The following application objects are used by the `interconnect_bill_of_material` UoF:

- `Alternate_product`;
- `Alternate_select_product`;
- `Assembly_alternate_product`;
- `Assembly_composition_relationship`;

- Assembly_material_composition_relationship;
- Interconnect_module_stratum_assembly_relationship;
- Material_composition_relationship;
- Next_higher_assembly_relationship;
- Physical_component_terminal;
- Promissory_usage_in_product_model_relationship;
- Promissory_usage_relationship;
- Substitute_product;
- Test_select_product.

4.1.54 interconnect_component_grouping

The interconnect_component_grouping UoF defines the information needed to describe the different ways of grouping components in an interconnect substrate. The interconnect_component_grouping UoF functionality supports fabricated assemblies. The interconnect_component_grouping UoF functionality supports design reuse of complex design elements.

The following application objects are used by the interconnect_component_grouping UoF:

- Array_placement_group;
- Assembly_component_placement_link;
- Assembly_group_component;
- Interfaced_group_component;
- Linear_array_placement_group_component;
- Linear_array_placement_group_component_link;
- Physical_component_terminal;
- Rectangular_array_placement_group_component;
- Sub_assembly_relationship.

4.1.55 `interconnect_design_information_assignment_to_usage_view`

The `interconnect_design_information_assignment_to_usage_view` UoF defines the information that describes the association between product data necessary for interconnect design and product data necessary to define a usage view of the interconnect.

The following application objects are used by the `interconnect_design_information_assignment_to_usage_view` UoF:

- `Component_feature_to_physical_usage_view_assignment`;
- `Component_terminal_to_interconnect_module_interface_terminal_assignment`;
- `Component_terminal_to_interconnect_module_join_terminal_assignment`;
- `Physical_component_terminal`;
- `Stratum_concept_to_physical_usage_view_assignment`.

4.1.56 `interconnect_functional_requirement_allocation`

The `interconnect_functional_requirement_allocation` UoF describes the assignment of functional connectivity requirements to an interconnect product under design. The assembly or assemblies for which the interconnect is originally designed to help compose constitutes an optional part of the requirements for the interconnect. The `Part_connected_terminals_definition` constitutes a definition of the required connections as seen from the exterior of the interconnect. The `Physical_node_requirement_to_implementing_component_allocation` is for use with `Printed_components` wherein the total functionality of the interconnect includes complex functionality (e.g., printed filters).

The following application objects are used by the `interconnect_functional_requirement_allocation` UoF:

- `Assembly_module`;
- `Connectivity_allocation_to_physical_network`;
- `Interconnect_module`;
- `Part_connected_terminals_definition`;
- `Physical_component_terminal`;
- `Physical_node_requirement_to_implementing_component_allocation`;
- `Test_requirement_allocation`.

4.1.57 **interconnect_interface_component**

The `interconnect_interface_component` UoF defines the information needed to describe the component that acts as a connector that provides external electrical access to an interconnect. The items' position in the interconnect substrate, and the specification that dictates the selection and placement are included.

The following application objects are used by the `interconnect_interface_component` UoF:

- `Ee_specification`;
- `Physical_component_terminal`;
- `Printed_connector_component`.

4.1.58 **interconnect_marking**

The `interconnect_marking` UoF defines the information needed to apply identification information, using material, to an interconnect. The content of the information is under configuration control. The scope of this UoF is for interconnect definitions.

The following application objects are used by the `interconnect_marking` UoF:

- `Configuration_controlled_message`;
- `Ee_specification`;
- `Marking`.

4.1.59 **interconnect_module_connection_routing**

The `interconnect_module_connection_routing` UoF defines the information needed to describe the electrical design requirements for the conductive elements associated with an interconnect substrate. The information contains specific topological requirements data, and user defined property data related to elements associated with the topological requirements data. The information needed to describe the resulting design is included. The `Physical_connectivity_definition` is in the scope of the `Physical_unit_network_definition` so as to not require a dummy `Assembly_module`.

EXAMPLE 1 Geometric trace layout data arranged in ordered trees is this type of information. Note that the context for the tree data is the assembly context and not the internal layout. The mapping between the two is accomplished with the `Junction_vertex_allocation` AO.

The following application objects are used by the `interconnect_module_connection_routing` UoF:

- `Conductive_interconnect_element_terminal_link`;
- `Connection_zone_based_fabrication_joint`;
- `Dependently_located_layer_connection_point`;
- `Dielectric_crossover_area`;
- `Explicitly_located_layer_connection_point`;
- `Fabrication_joint`;
- `Inter_stratum_join_implementation`;
- `Inter_stratum_join_relationship`;
- `Intra_stratum_join_relationship`;
- `Join_relationship`;
- `Junction_vertex_allocation`;
- `Layer_connection_point`;
- `Layer_connection_point_link`;
- `Partially_routed_physical_network`;
- `Physical_component_terminal`;
- `Physical_connectivity_abstraction_map`;
- `Physical_connectivity_definition`;
- `Physical_connectivity_definition_relationship`;
- `Physical_connectivity_element`;
- `Physical_connectivity_structure_definition`;
- `Physical_network`;
- `Physical_network_supporting_inter_stratum_feature`;

- Physical_network_supporting_stratum_feature_conductive_join;
- Physical_unit_network_definition;
- Stratum_concept_relationship;
- Stratum_feature_conductive_join;
- Topological_junction;
- Unrouted_join_relationship.

4.1.60 interconnect_module_design

The interconnect_module_design UoF defines the information needed to describe the required physical design data for an interconnect substrate. The information includes stackup, conductive patterns, feature tolerances, design intent, interface feature design, plated and unplated passages, various forms of cutout and edge treatments, and traceability capability from design features back to design intent. Various design definitions may exist for the same interconnect module depending on the discipline and enterprise perspective.

The following application objects are used by the interconnect_module_design UoF:

- Adjacent_stratum_surface_definition;
- Adjacent_stratum_surface_embedded_component_surface_definition;
- Array_placement_group;
- Assembly_component;
- Assembly_composition_relationship;
- Assembly_group_component;
- Blind_via;
- Bonded_conductive_base_blind_via;
- Buried_via;
- Component_based_annotation_symbol_placement_relationship;
- Component_based_annotation_text_relationship;
- Component_external_reference;

- Component_feature;
- Component_feature_external_reference;
- Component_feature_make_from_relationship;
- Component_feature_relationship;
- Component_feature_to_layout_feature_relationship;
- Component_interface_terminal;
- Component_make_from_relationship;
- Component_terminal;
- Component_termination_passage;
- Component_termination_passage_interface_terminal;
- Component_termination_passage_join_terminal;
- Conductive_filled_area;
- Conductive_interconnect_element;
- Conductive_interconnect_element_terminal;
- Conductive_interconnect_element_with_pre_defined_transitions;
- Conductive_interconnect_element_with_user_defined_single_transition;
- Conductor;
- Connected_area_component;
- Connected_filled_area;
- Connection_zone_based_fabrication_joint;
- Cutout;
- Cutout_edge_segment;
- Design_layer_stratum;

- Dielectric_material_passage;
- Documentation_layer_stratum;
- Edge_segment_vertex;
- Electrical_isolation_laminate_component;
- Embedded_component_terminal;
- Fabrication_joint;
- Fiducial;
- Fiducial_stratum_feature;
- Filled_area_material_removal_laminate_component;
- Footprint_occurrence;
- Footprint_occurrence_sub_assembly_relationship;
- Frozen_assignment;
- Inter_stratum_extent;
- Inter_stratum_feature;
- Inter_stratum_feature_edge_segment_template;
- Inter_stratum_feature_edge_template;
- Interconnect_module;
- Interconnect_module_edge;
- Interconnect_module_edge_segment;
- Interconnect_module_macro_component;
- Interconnect_module_macro_component_join_terminal;
- Interconnect_module_stratum_assembly_relationship;
- Interconnect_module_stratum_based_terminal;

- Interface_access_material_removal_laminate_component;
- Interface_access_stratum_feature_template_component;
- Interfacial_connection;
- Internal_probe_access_area;
- Internal_stratum_access;
- Join_two_physical_connectivity_definition_supporting_inter_stratum_feature;
- Join_two_physical_connectivity_definition_supporting_printed_component;
- Laminate_component;
- Laminate_component_feature;
- Laminate_component_interface_terminal;
- Laminate_component_join_terminal;
- Laminate_group_component_make_from_relationship;
- Laminate_text_component;
- Laminate_text_string_component;
- Land_interface_terminal;
- Land_join_terminal;
- Layer;
- Layout_macro_component;
- Layout_macro_definition;
- Layout_macro_definition_terminal_to_usage_terminal_assignment;
- Layout_macro_floor_plan_template;
- Material_removal_laminate_component;
- Multi_layer_material_removal_laminate_component;

- Multi_layer_stratum_feature;
- Multi_layer_stratum_feature_template_component;
- Next_higher_assembly_relationship;
- Non_conductive_base_blind_via;
- Non_functional_land_interface_terminal;
- Non_functional_land_join_terminal;
- Padstack_occurrence;
- Padstack_occurrence_sub_assembly_relationship;
- Part_connected_terminals_definition;
- Part_template;
- Partially_plated_cutout;
- Partially_plated_interconnect_module_edge;
- Pcb;
- Physical_component_terminal;
- Physical_connectivity_interrupting_cutout;
- Physical_connectivity_make_from_relationship;
- Plated_conductive_base_blind_via;
- Plated_cutout;
- Plated_cutout_edge_segment;
- Plated_inter_stratum_feature;
- Plated_interconnect_module_edge;
- Plated_interconnect_module_edge_segment;
- Plated_passage;

- Primary_stratum_indicator_symbol;
- Printed_component;
- Printed_component_join_terminal;
- Printed_component_link;
- Printed_connector_component;
- Printed_connector_component_interface_terminal;
- Probe_access_area;
- Promissory_usage_in_product_model_relationship;
- Promissory_usage_relationship;
- Routed_printed_component;
- Shield;
- Special_symbol_laminate_component;
- Stratum;
- Stratum_average_surface;
- Stratum_feature;
- Stratum_feature_in_stratum_3d_position;
- Stratum_feature_in_stratum_planar_position;
- Stratum_feature_template_component;
- Stratum_interconnect_module_3d_position;
- Stratum_interconnect_module_planar_position;
- Stratum_make_from_relationship;
- Stratum_surface;
- Sub_assembly_relationship;

- Thermal_component;
- Thermal_isolation_laminate_component;
- Unrouted_conductive_interconnect_element;
- Unsupported_passage;
- Via;
- Via_terminal.

4.1.61 interconnect_module_edge

The interconnect_module_edge UoF defines the information needed to describe the outer edge of the interconnect substrate in a design view.

The following application objects are used by the interconnect_module_edge UoF:

- Edge_segment_vertex;
- Interconnect_module_edge;
- Interconnect_module_edge_segment;
- Partially_plated_interconnect_module_edge;
- Physical_component_terminal;
- Plated_interconnect_module_edge;
- Plated_interconnect_module_edge_segment.

4.1.62 interconnect_module_usage_view

The interconnect_module_usage_view UoF defines the information needed to describe the interface definition of the interconnect substrate. This view may be used for the purpose of simulation and verifying the mechanical fit of the interconnect substrate when installed in the next level of assembly.

The following application objects are used by the interconnect_module_usage_view UoF:

- Edge_segment_vertex;
- Interconnect_module_cavity_surface;

- Interconnect_module_cutout_segment_surface;
- Interconnect_module_cutout_surface;
- Interconnect_module_edge_segment_surface;
- Interconnect_module_edge_surface;
- Interconnect_module_interface_terminal;
- Interconnect_module_join_terminal;
- Interconnect_module_primary_surface;
- Interconnect_module_secondary_surface;
- Interconnect_module_stratum_based_terminal;
- Interconnect_module_surface_feature;
- Interconnect_module_terminal;
- Interconnect_module_terminal_surface_constituent_relationship;
- Interconnect_module_usage_view;
- Located_interconnect_module_thickness_requirement;
- Pcb_usage_view.

4.1.63 interconnect_physical_requirement_allocation

The interconnect_physical_requirement_allocation UoF describes the assignment of customer, regulatory agency, and enterprise requirements to an interconnect product under design. These requirements may be assigned to physical items, geometric features, or conceptual items that are used in various phases of the design process.

The following application objects are used by the interconnect_physical_requirement_allocation UoF:

- Assembly_requirement_allocation;
- Assembly_separation_requirement_allocation;
- Electromagnetic_compatibility_requirement_allocation;
- Layout_land_width_tolerance_requirement_occurrence;

- Layout_line_width_tolerance_requirement_occurrence;
- Physical_component_terminal;
- Requirement_allocation;
- Shield_allocation;
- Specification_allocation;
- Thermal_isolation_requirement_assignment;
- Thermal_requirement_allocation.

4.1.64 interconnect_placement_requirements

The interconnect_placement_requirements UoF defines the information needed to describe the constraints restricting where components are to be placed in an Interconnect_module. These restrictions may be component category to board feature category or component category to component category or board feature category to board feature category related. Data that refers to assembly is included in order to support embedded components only, where those components are assembled during the substrate fabrication process.

The following application objects are used by the interconnect_placement_requirements UoF:

- Component_external_reference;
- Component_feature_external_reference;
- Component_group_assignment;
- Component_placement_restriction_assignment;
- Interconnect_module_component_surface_feature;
- Interconnect_module_constraint_region;
- Item_restricted_requirement_occurrence;
- Layer_qualified_layout_spacing_requirement_occurrence;
- Layout_spacing_requirement_non_conforming_design_object;
- Layout_spacing_requirement_occurrence;
- Mounting_restriction_area;

- Mounting_restriction_volume;
- Physical_component_terminal;
- Placement_group;
- Placement_group_area_assignment;
- Placement_group_volume_assignment.

4.1.65 interconnect_shape

The interconnect_shape UoF defines the information needed to describe explicit geometric models used in the design process for an interconnect substrate. This UoF permits the type of geometric representation for a library item to be different than the type of geometric representation of an instance in the design.

The following application objects are used by the interconnect_shape UoF:

- Assembly_component_2d_shape;
- Assembly_component_3d_shape;
- Assembly_component_non_planar_shape;
- Bond_assembly_position;
- Component_2d_edge_location;
- Component_2d_embedded_location;
- Component_2d_location;
- Component_3d_location;
- Component_assembly_2d_position;
- Component_assembly_3d_position;
- Component_assembly_non_planar_position;
- Component_non_planar_location;
- Component_part_2d_geometric_representation_relationship;
- Component_part_2d_non_planar_geometric_representation_relationship;

- Component_part_3d_geometric_representation_relationship;
- Design_intent_modification_3d_position;
- Design_intent_modification_3d_shape;
- Design_intent_modification_non_planar_2d_shape;
- Design_intent_modification_planar_position;
- Design_intent_modification_planar_shape;
- Design_intent_modification_styled_area_representation;
- Design_view_connection_zone_position_relationship;
- Design_view_terminal_component_shape_relationship;
- Open_shell_based_surface;
- Physical_component_terminal;
- Stratum_3d_shape;
- Stratum_average_surface_shape;
- Stratum_feature_3d_shape;
- Stratum_feature_non_planar_2d_shape;
- Stratum_feature_planar_shape;
- Stratum_feature_shape_stratum_average_surface_shape_relationship;
- Stratum_feature_shape_stratum_surface_shape_relationship;
- Stratum_feature_styled_area_representation;
- Stratum_planar_shape;
- Stratum_surface_in_stratum_3d_position;
- Stratum_surface_shape.

4.1.66 laminate_assembly_design

The laminate_assembly_design UoF defines the information needed to describe the laminate product resulting from the adjacent surface to surface bonding between laminate components. The usage of Assembly_joint in this UoF is to allow specification of the sequential arrangement of materials.

EXAMPLE 1 This design data may be used during a concurrent engineering process where one side is viewing the design as an arrangement of Stratum in the entire interconnect design and the other side is viewing the design as an arrangement of laminates.

The following application objects are used by the laminate_assembly_design UoF:

- Assembly_joint;
- Component_mating_constraint_condition;
- Interconnect_module;
- Interconnect_module_component;
- Interconnect_module_component_surface_feature;
- Next_higher_assembly_relationship;
- Physical_component_terminal;
- Promissory_usage_in_product_model_relationship;
- Promissory_usage_relationship.

4.1.67 land

The land UoF defines the information required to partially specify certain interconnect substrate features. These items are defined to support attachments for mating components in the electronic assembly. The items are not necessarily themselves visually distinguishable in the material in the interconnect substrate.

The following application objects are used by the land UoF:

- Component_termination_passage_and_contact_size_dependent_land;
- Component_termination_passage_and_contact_size_dependent_non_functional_land;
- Component_termination_passage_dependent_land;
- Component_termination_passage_dependent_non_functional_land;

- Contact_size_dependent_land;
- Contact_size_dependent_non_functional_land;
- Land;
- Non_functional_land;
- Unsupported_passage_dependent_non_functional_land;
- Via_and_contact_size_dependent_land;
- Via_and_contact_size_dependent_non_functional_land;
- Via_dependent_land;
- Via_dependent_non_functional_land.

4.1.68 library_item

The library_item UoF defines the information needed to describe the generic items that may be configuration managed at a level appropriate for inclusion in a library. Items in a library are incorporated into many designs without actively tracking which designs incorporate the items. These items are readily available to any enterprise; however, the amount of descriptive information varies between enterprises. The property values may vary between enterprises.

The following application objects are used by the library_item UoF:

- Library_ee_product_definition.

4.1.69 manufacturing_technology

The manufacturing_technology UoF defines the information needed to describe the properties of the fabrication technology selected for the interconnect substrate being designed.

The following application objects are used by the manufacturing_technology UoF:

- Component_termination_passage_template;
- Counterbore_passage_template;
- Countersunk_passage_template;
- Default_attachment_size_and_component_termination_passage_based_land_physical_template;

- Default_attachment_size_and_via_based_land_physical_template;
- Default_attachment_size_based_land_physical_template;
- Default_component_termination_passage_based_land_physical_template;
- Default_component_termination_passage_definition;
- Default_land_physical_template;
- Default_passage_based_land_physical_template;
- Default_unsupported_passage_based_land_physical_template;
- Default_unsupported_passage_definition;
- Default_via_based_land_physical_template;
- Default_via_definition;
- Dependent_material_removal_feature_template;
- Design_layer_technology;
- Documentation_layer_technology;
- Fabrication_technology_specification;
- Inter_stratum_feature_template;
- Land_physical_template;
- Land_template_interface_terminal;
- Land_template_join_terminal;
- Land_template_terminal;
- Line_width_class;
- Passage_technology;
- Stratum_technology;
- Unsupported_passage_template;
- Via_template.

4.1.70 material_product

The material_product UoF defines the information needed to describe the materials that compose an item. The association between specific materials used during the manufacturing process of the interconnect substrate are also described, in addition to any components that are part of the interconnect substrate manufacturing process. Material product categories may be described in addition to material properties. Bulk material callout is supported, as well as generic callout on engineering design definitions.

NOTE To form a material assembly, this edition of the standard relies on Assembly_composition_relationship but since this UoF does not exclusively support a Conformance Class and Conformance Classes that reference this UoF already call out that Application Object, that Application Object is not included in this UoF.

The following application objects are used by the material_product UoF:

- Ee_material;
- Ee_product_category;
- Ee_product_category_relationship;
- Ee_product_related_product_category;
- Material_composition_relationship;
- Material_definition.

4.1.71 non_topological_surface

The non_topological_surface UoF describes the representation of the shape of a product using surface geometry without topology. These representations are formed only by the use of points, curves, and surfaces. The boundaries of the surfaces are defined by curves on the surfaces and by explicit association between curves and surfaces they bound. Surfaces and curves shall be explicitly trimmed unless they are closed.

The following application objects are used by the non_topological_surface UoF:

- Geometrically_bounded_surface.

4.1.72 package

The package UoF defines the information needed to describe the package of a component that is part of the assembly. The shape of the component both as purchased and as delivered to a customer are provided. A common identifier for the shape of the package and placement orientation alignment information between 2 dimensional and 3 dimensional models are also provided. The information needed to allow industry or enterprise groups to populate explicit models and define associated properties and associated categories for standard packages and their versions is provided.

The following application objects are used by the package UoF:

- Altered_package;
- Altered_package_terminal;
- Connection_zone_interface_plane_relationship;
- Fiducial_part_feature;
- Guided_wave_terminal;
- Length_trimmed_terminal;
- Package;
- Package_body;
- Package_body_bottom_surface;
- Package_body_edge_segment_surface;
- Package_body_edge_surface;
- Package_body_surface;
- Package_body_top_surface;
- Package_terminal;
- Package_terminal_surface_constituent_relationship;
- Part_tooling_feature;
- Polarity_indication_feature;
- Primary_orientation_feature;
- Primary_reference_terminal;
- Seating_plane;
- Secondary_orientation_feature;
- Shape_formed_terminal;

- Surface_prepped_terminal;
- Tertiary_orientation_feature;
- Test_point_part_feature;
- Wire_terminal.

4.1.73 packaged_part_black_box_model

The packaged_part_black_box_model UoF defines the information necessary to describe a product as a part for the next level of assembly. The inner structure of this part isn't visible by the electronic assembly designer. This UoF defines the information necessary to describe the product terminals that are directly connected for the purposes of normal usage.

EXAMPLE 1 All terminals that are to be connected to a certain power supply voltage may be specified by use of the information defined in this UoF.

NOTE When an enterprise designs a part specifically for use on an electronic assembly, this allows the designer of the electronic assembly to have just enough information to use that part, like any other part, in the design process of the electronic assembly.

The following application objects are used by the packaged_part_black_box_model UoF:

- Altered_packaged_part;
- Packaged_connector;
- Packaged_connector_terminal_relationship;
- Packaged_part;
- Packaged_part_interface_terminal;
- Packaged_part_join_terminal;
- Packaged_part_terminal;
- Part_connected_terminals_definition.

4.1.74 packaged_part_terminal_analytical_representation

The packaged_part_terminal_analytical_representation UoF defines the information needed to associate unique behavioural data for each specific terminal to support computer-based experimentation that explicitly includes terminal effects. The specific context of this UoF is that the individual terminal is treated as a separate model for analysis purposes. Each Application object type referenced in this UoF is allocating the analysis port to the external feature or is allocating the analysis port to the internal feature. Each discipline may have a different analytical representation of a product design. For higher fidelity, reference the connection_zone_to_analytical_model_port_mapping UoF or a datum_difference related UoF since those UoFs more detailed modeling capabilities.

NOTE This UoF is only appropriate in the context of a so-called white box model.

The following application objects are used by the packaged_part_terminal_analytical_representation UoF:

- Part_device_terminal_model_port_assignment;
- Part_feature_model_port_assignment.

4.1.75 packaged_part_white_box_model

The packaged_part_white_box_model UoF defines the information necessary to describe a product for use by analysts for evaluation of properties as part of the next level of assembly. Some portion of the inner structure of the part is visible by analysts. Users may define properties at a level of accuracy required to support enterprise or standard analysis requirements.

NOTE This capability is typically used where the packaged parts are complex and are also primary sources of assembly failures. This information is not a description of multi-level hierarchical design definitions. This information may be used where off-the-shelf products are used where only critical information is conveyed.

EXAMPLE 1 The use of this information for analyzing purchased MCMs is typical.

The following application objects are used by the packaged_part_white_box_model UoF:

- Connection_zone_interface_plane_relationship;
- Connection_zone_model_port_assignment;
- Device_2d_position;
- Device_3d_position;
- Device_terminal_map;
- Device_terminal_map_relationship;

- Packaged_connector_terminal_relationship;
- Packaged_part;
- Packaged_part_interface_terminal;
- Packaged_part_join_terminal;
- Part_connected_terminals_definition;
- Part_device;
- Part_device_terminal;
- Part_device_terminal_model_port_assignment;
- Part_feature_model_port_assignment;
- Usage_view_connection_zone_terminal_shape_relationship.

4.1.76 part_feature

The part_feature UoF defines the information needed to describe the distinctive aspects of separately manufactured products included in the design. Explicit models of the aspects may be combined in the explicit model for the products. Classification of feature functionality is supported, with several predefined functions provided.

The following application objects are used by the part_feature UoF:

- Edge_segment_vertex;
- Feature_group_relationship;
- Fiducial_part_feature;
- Interconnect_module_cavity_surface;
- Interconnect_module_cutout_segment_surface;
- Interconnect_module_cutout_surface;
- Interconnect_module_edge_segment_surface;
- Interconnect_module_edge_surface;
- Interconnect_module_primary_surface;

- Interconnect_module_secondary_surface;
- Interconnect_module_surface_feature;
- Interconnect_module_terminal_surface_constituent_relationship;
- Part_array_feature;
- Part_feature;
- Part_feature_placement_link;
- Part_group_feature;
- Part_interface_access_feature;
- Part_linear_array_feature;
- Part_linear_array_feature_link;
- Part_mating_feature;
- Part_mounting_feature;
- Part_rectangular_array_feature;
- Part_terminal;
- Part_tooling_feature;
- Polarity_indication_feature;
- Primary_orientation_feature;
- Primary_reference_terminal;
- Secondary_orientation_feature;
- Tertiary_orientation_feature;
- Test_point_part_feature;
- Thermal_feature;
- Usage_concept_usage_relationship;
- Usage_view_connection_zone_terminal_shape_relationship.

4.1.77 **part_feature_to_analytical_model_port_mapping**

The `part_feature_to_analytical_model_port_assignment` UoF defines the information needed to associate a specific feature of a physical unit with an analytical model port to support computer-based experimentation. The consequence is that the UoF defines the information needed to associated a physical unit with a specific behaviour.

The following application objects are used by the `part_feature_to_analytical_model_port_mapping` UoF:

- `Analytical_representation`;
- `Part_feature_model_port_assignment`.

4.1.78 **part_template**

The `part_template` UoF defines the information necessary to describe the different types of templates. These templates provide the shape and other property information for the feature to be used in the design of the interconnect substrate. These templates reference a `Template_definition` for the purposes of identification.

The following application objects are used by the `part_template` UoF:

- `Connection_zone_map_identification`;
- `Default_trace_template`;
- `Design_layer_type_specific_padstack_definition`;
- `Electrical_isolation_removal_template`;
- `Footprint_definition`;
- `Layout_macro_definition_terminal_to_usage_terminal_assignment`;
- `Layout_macro_floor_plan_template`;
- `Material_addition_feature_template`;
- `Material_removal_feature_template`;
- `Padstack_definition`;
- `Padstack_location_in_footprint_definition`;
- `Part_template`;

- Part_template_location_in_padstack_definition;
- Part_text_template;
- Special_symbol_part_template;
- Stratum_feature_template;
- Stratum_occurrence_specific_padstack_definition;
- Stratum_type_independent_padstack_definition;
- Template_definition;
- Template_stratum_assignment;
- Thermal_isolation_removal_template;
- Trace_template;
- Usage_concept_usage_relationship.

4.1.79 part_template_shape

The part_template_shape UoF defines the information needed to describe explicit geometric models used as templates in the design process for an electronic assembly or for an interconnect substrate. This UoF defines the information needed to associate the 2 dimensional and 3 dimensional geometric representation of the layout templates in order to state that they are consistent.

The following application objects are used by the part_template_shape UoF:

- Footprint_definition_shape;
- Padstack_definition_shape;
- Padstack_location_in_footprint_definition_transform;
- Part_template_3d_keepout_shape;
- Part_template_3d_shape;
- Part_template_location_in_padstack_definition_transform;
- Part_template_non_planar_2d_shape;
- Part_template_or_physical_unit_2d_shape_representation_relationship;

- Part_template_or_physical_unit_3d_shape_representation_relationship;
- Part_template_planar_keepout_shape;
- Part_template_planar_shape.

4.1.80 physical_network

The physical_network UoF defines the information needed to describe the internal topological definition used as an explicit requirement in the design of the interconnect substrate. This internal topological definition is a result of mapping the external connection requirements defined in the interconnect_functional_requirement_allocation UoF to the interconnect design view.

The following application objects are used by the physical_network UoF:

- Partially_routed_physical_network;
- Physical_component_terminal;
- Physical_network.

4.1.81 physical_unit

The physical_unit UoF defines the information necessary to describe a product that may be an electronic assembly, or be used in an electronic assembly, or be used in the design of an electronic assembly. The physical_unit UoF defines the information necessary to describe a product that may be an interconnect product, or be used in an interconnect product, or be used in the design of an interconnect product. The information shall at least describe how the product may be used as a component but, may also encompass all the information required to reproduce the item. The information may also include information necessary to categorize the product or product versions for part selection activities. The information may also include information for parts that are purely mechanical in function.

The following application objects are used by the physical_unit UoF:

- Ee_product_category;
- Ee_product_category_relationship;
- Ee_product_related_product_category;
- Ee_product_specific_parameter_value_assignment;
- Physical_unit;
- Physical_unit_design_view;

- Physical_unit_relationship;
- Physical_unit_usage_view;
- Technology;
- Usage_concept_usage_relationship.

4.1.82 physical_unit_analytical_representation

The physical_unit_analytical_representation UoF defines the information needed to associate specific data for a physical unit with an analytical model to support computer-based experimentation. Each discipline may have a different physical unit analytical representation. Direct specification of the mapping to a connection area on the terminal of the product concepts under investigation is provided.

The following application objects are used by the physical_unit_analytical_representation UoF:

- Analytical_representation;
- Connection_zone_model_port_assignment;
- Part_feature_model_port_assignment.

4.1.83 physical_unit_shape

The physical_unit_shape UoF defines the information needed to describe explicit geometric models used in the design process for an electronic assembly or for an interconnect substrate. This UoF permits different types of geometric representation for the shape of the physical units and synchronization of the shapes with different dimensionality.

The following application objects are used by the physical_unit_shape UoF:

- Part_template_or_physical_unit_2d_shape_representation_relationship;
- Part_template_or_physical_unit_3d_shape_representation_relationship;
- Physical_unit_3d_keepout_shape;
- Physical_unit_3d_shape;
- Physical_unit_planar_keepout_shape;
- Physical_unit_planar_shape.

4.1.84 **planned_characteristic**

The `planned_characteristic` UoF defines the information that captures design intent properties. This design intent is based upon the manufacturing capabilities, product design structure, and component characteristics. This UoF defines the information that captures properties that are the result of experimental evaluation of the product definition in the product's intended use or in a manufacturing environment.

The following application objects are used by the `planned_characteristic` UoF:

- `Evaluated_characteristic`;
- `Evaluation_ee_product_definition`;
- `Planned_characteristic`;
- `Rule_based_evaluated_characteristic`.

4.1.85 **planned_effectivity**

The `planned_effectivity` UoF defines the information needed to describe under what circumstances a configuration change is to occur. The effectivity may be based on either the manufacturing cycle, or a time period, or a given quantity, or a range of individual products.

EXAMPLE 1 A planned-effectivity would be the time period between two dates.

The following application objects are used by the `planned_effectivity` UoF:

- `Planned_date_effectivity`;
- `Planned_effectivity`;
- `Planned_lot_effectivity`;
- `Planned_serial_number_effectivity`.

4.1.86 **printed_part_template**

The `printed_part_template` UoF defines the information needed to describe the definition of a printed component. Support is provided for both microwave and non-microwave design. Functionality is assigned in the definition. This UoF defines the information needed to describe the terminals that are directly connected internally in the printed component.

The following application objects are used by the printed_part_template UoF:

- Non_conductive_cross_section_template;
- Physical_component_terminal;
- Printed_connector_template;
- Printed_connector_template_terminal_relationship;
- Printed_part_cross_section_template;
- Printed_part_cross_section_template_terminal;
- Printed_part_template;
- Printed_part_template_connected_terminals_definition;
- Printed_part_template_interface_terminal;
- Printed_part_template_join_terminal;
- Printed_part_template_link;
- Printed_part_template_material;
- Printed_part_template_material_link;
- Printed_part_template_terminal;
- Printed_part_template_terminal_relationship;
- Stratum_technology_link;
- Template_material_cross_section_boundary;
- Usage_concept_usage_relationship.

4.1.87 product_identification

The product_identification UoF describes the structure through which a product, product categories, unique versions of a product, and different design discipline views of a product may be defined. This part of ISO 10303 does not rely solely on the relationship between design discipline view of a product and the product version to establish relationships between design discipline views but provides explicit models of the desired relationships in order to meet domain requirements(e.g., one functional product model may define the functionality of many physical products).

The following application objects are used by the product_identification UoF:

- Ee_product;
- Ee_product_category;
- Ee_product_category_relationship;
- Ee_product_definition;
- Ee_product_related_product_category;
- Ee_product_specific_parameter_value_assignment;
- Ee_product_version;
- Ee_product_version_supersedence;
- Externally_defined_ee_product_definition;
- Product_association.

4.1.88 product_requirement_allocation

The product_requirement_allocation UoF describes the assignment of customer, regulatory agency, and enterprise requirements to a product. These requirements may be assigned to products or configuration items or product versions in various phases of the design process.

The following application objects are used by the product_requirement_allocation UoF:

- Requirement_allocation.

4.1.89 product_rule

The product_rule UoF defines the information needed to describe design constraints, commonly referred to as rules. The information describes the rule's relationship to product characteristics, formal description of the functions that embody the rule, in addition to the configuration management aspects applied to the rule. The configuration management information supports rule version control and supports a formal process to making changes to an existing rule. A rule is an Ee_product.

The following application objects are used by the product_rule UoF:

- Ee_product;
- Ee_rule_complex_clause;

- Ee_rule_conclusion_definition;
- Ee_rule_definition;
- Ee_rule_function;
- Ee_rule_premise_definition;
- Ee_rule_simple_clause;
- Ee_rule_version;
- Participant_date_and_time_assignment;
- Rule_action;
- Rule_adoption;
- Rule_change_request;
- Rule_creation;
- Rule_expiration;
- Rule_justification;
- Rule_modification;
- Rule_priority;
- Rule_rejection;
- Rule_request;
- Rule_set;
- Rule_set_group;
- Rule_submission;
- Rule_supersedence.

4.1.90 requirement_decomposition

The requirement_decomposition UoF defines the information needed to describe the structure of a requirement, including any explicit combinatorial relationship between requirements helping to compose a requirement, and any dependencies upon other requirements.

The following application objects are used by the requirement_decomposition UoF:

- Constraint_occurrence;
- Design_characteristic_occurrence;
- Design_requirement_occurrence;
- Ee_product_definition_based_constraint_occurrence;
- Ee_requirement_occurrence;
- Electrical_requirement_occurrence;
- Electromagnetic_requirement_occurrence;
- Fabrication_thickness_requirement_occurrence;
- Inspection_requirement_occurrence;
- Layout_land_width_tolerance_requirement_occurrence;
- Layout_line_width_tolerance_requirement_occurrence;
- Material_electrical_conductivity_requirement_occurrence;
- Mechanical_requirement_occurrence;
- Operational_requirement_occurrence_relationship;
- Purpose_occurrence;
- Requirement_composition;
- Requirement_definition;
- Requirement_group_occurrence;
- Rule_based_constraint_occurrence;

- Shape_and_ee_product_definition_based_constraint_occurrence;
- Test_requirement_occurrence;
- Thermal_requirement_occurrence.

4.1.91 sequential_laminate_assembly_design

The sequential_laminate_assembly_design UoF defines the information needed to describe the laminate product resulting from the adjacent surface to surface bonding between laminate components. The sequential_laminate_assembly_design UoF also defines the information needed to describe a multi-layer connection in the laminate assembly where the layers are not adjacent. This description uses Interconnect_module, Interconnect_module_component_stratum_based_terminal, and Interconnect_module_product_assembly_view_fabrication_joint.

EXAMPLE 1 This design data may be used during a concurrent engineering process where one side is viewing the design as an arrangement of Stratum in the entire interconnect design and the other side is viewing the design as an arrangement of laminates.

EXAMPLE 2 This design data may be used during a concurrent engineering process where one side is viewing the design as the entire interconnect substrate, and the other side is viewing only the data required to determine the connections at one level of laminate assembly. Since the bonds in the laminate assembly are associated with occurrences of Interconnect_module_stratum_based_terminal, the enterprise responsible for the entire design may populate Part_connected_terminals_definition from the functional_specification UoF to define to the lower level design team the required connectivity for the laminate assembly at the lower level.

NOTE Fabrication technology capability will limit the number of lower level bonds actually associated to each Interconnect_module_product_assembly_view_fabrication_joint, but this part of ISO 10303 does not preclude any specific fabrication technology. Cutout and other Inter_stratum_feature are not within the scope of the sequential_laminate_assembly_design UoF as they directly depend on Stratum which is not included in this UoF.

The following application objects are used by the sequential_laminate_assembly_design UoF:

- Assembly_material_composition_relationship;
- Interconnect_module;
- Interconnect_module_component;
- Interconnect_module_component_stratum_based_terminal;
- Interconnect_module_component_surface_feature;
- Interconnect_module_product_assembly_view_fabrication_joint;
- Interconnect_module_stratum_based_terminal;

- Next_higher_assembly_relationship;
- Physical_component_terminal;
- Promissory_usage_in_product_model_relationship;
- Promissory_usage_relationship.

4.1.92 shape

The shape UoF defines the information needed to describe explicit geometric models used in the design process for an electronic assembly or for an interconnect substrate.

The following application objects are used by the shape UoF:

- Axis_placement;
- Axis_placement_2d;
- Axis_placement_3d;
- Boolean_csg_result;
- Bound_volume_shape;
- Bounding_curve;
- Cartesian_coordinate_system;
- Cartesian_point;
- Cartesian_transformation_operator_2d;
- Circular_area;
- Closed_curve;
- Composite_curve;
- Composite_curve_segment;
- Connection_zone;
- Csg_2d_shape;
- Curve;

- Direction;
- Elliptic_area;
- Externally_defined_hatch_style;
- Externally_defined_tile;
- Externally_defined_tile_style;
- Feature_shape_definition;
- Fill_area;
- Fill_area_occurrence;
- Fill_area_style;
- Fill_area_style_hatching;
- Fill_area_style_tile;
- Fill_area_style_tile_curve_with_style;
- Fill_area_style_tile_solid_region;
- Fill_area_style_tile_symbol_with_style;
- Geometric_representation_relationship;
- Manifold_subsurface_shape;
- Non_feature_shape_definition;
- Planar_projected_shape;
- Planar_shape;
- Polygonal_area;
- Pre_defined_datum_axis_3d_symbol;
- Pre_defined_datum_plane_3d_symbol;
- Pre_defined_parallel_datum_axis_2d_symbol;

- Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship;
- Pre_defined_perpendicular_datum_axis_2d_symbol;
- Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship;
- Pre_defined_perpendicular_datum_plane_2d_symbol;
- Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship;
- Primitive_2d;
- Rectangular_area;
- Shape_composition_relationship;
- Shape_definition;
- Shape_item;
- Solid_fill_style;
- Trimmed_line;
- Vector;
- View_level_non_feature_shape_definition;
- Volume_shape_intersection;
- Volume_shape_plane_projection;
- Volume_shape_point_projection.

4.1.93 signal

The signal UoF defines the information needed to describe properties of an electrical physical phenomenon. This physical phenomenon property may be a simple instance or a composition of many simple instances.

The following application objects are used by the signal UoF:

- Characteristic_data_set;
- Characteristic_data_table;

- Composite_signal_property;
- Composite_signal_property_relationship;
- Signal;
- Signal_property.

4.1.94 solid_of_linear_extrusion

The solid_of_linear_extrusion UoF defines the information needed to describe a volumetric shape defined through the process of sweeping a planar surface over a specified distance.

The following application objects are used by the solid_of_linear_extrusion UoF:

- Solid_of_linear_extrusion.

4.1.95 source_control

The source_control UoF defines the information needed to describe an organization that is qualified to produce a particular product.

The following application objects are used by the source_control UoF:

- Certification;
- Organization;
- Supplied_product_version.

4.1.96 stratum

The stratum UoF defines the information needed to describe the individual layers of the interconnect substrate as they occur in the design definition. Various design definitions may exist for the same interconnect substrate, depending on the discipline and enterprise perspective.

The following application objects are used by the stratum UoF:

- Design_layer_stratum;
- Documentation_layer_stratum;
- Layer;
- Stratum;

- Stratum_average_surface;
- Stratum_surface;
- Stratum_technology.

4.1.97 stratum_feature

The stratum_feature UoF defines the information needed to describe the geometric shape of material features used to compose each stratum in an interconnect substrate. Location information is included in the description, in addition to the association to constraints placed upon the design. The stratum feature UoF defines the information needed to describe the sequentially ordered changes from an original or base-line feature design intent to a present feature design. Changes in feature material, either removal or addition, may be described, in addition to the nature of the requirement for the feature modification, either for electrical or thermal isolation. Design intent priority may be assigned to stratum features to constrain the extent of changes.

The following application objects are used by the stratum_feature UoF:

- Design_intent_modification;
- Design_intent_modification_3d_position;
- Design_intent_modification_planar_position;
- Design_intent_modification_sequence;
- Electrical_isolation_removal_structure;
- Filled_area_material_removal_structure;
- Material_addition_feature;
- Material_removal_feature;
- Material_shape_modification;
- Physical_component_terminal;
- Stratum_feature;
- Stratum_feature_in_stratum_3d_position;
- Stratum_feature_in_stratum_planar_position;
- Thermal_isolation_removal_structure.

4.1.98 stratum_position

The stratum_position UoF defines the information needed to describe where the stratum is located and its orientation with respect to the coordinate system of the interconnect substrate. The position supports either a three-dimensional coordinate system or a two-dimensional coordinate system.

The following application objects are used by the stratum_position UoF:

- Adjacent_stratum_surface_definition;
- Adjacent_stratum_surface_embedded_component_surface_definition;
- Interconnect_module_stratum_assembly_relationship;
- Stratum_interconnect_module_3d_position;
- Stratum_interconnect_module_planar_position.

4.1.99 styled_curve

The styled_curve UoF defines the information needed to describe the geometric properties of material as constrained by a shape. The shape information included is parametric, and path based, rather than an explicit closed curve model.

The following application objects are used by the styled_curve UoF:

- Curve_style;
- Curve_style_font;
- Curve_style_font_and_scaling;
- Externally_defined_curve_font;
- Solid_curve;
- Styled_curve_occurrence.

4.1.100 styled_text

The styled_text UoF defines the information needed to describe the appearance of material included in an interconnect substrate for conveying textual information to the reader.

The following application objects are used by the styled_text UoF:

- Font_definition;
- Registered_font;
- Styled_text.

4.1.101 test_probe

The test_probe UoF defines the information needed to describe the shape and location of a feature that is used to facilitate functional testing and analysis of an electronic assembly, or an interconnect substrate.

The following application objects are used by the test_probe UoF:

- Internal_probe_access_area;
- Probe_access_area.

4.1.102 wireframe_2d_model_with_topology

The wireframe_2d_model_with_topology UoF defines the information needed to describe the representation of the shape of a product by wireframes in 2 dimensions that are defined by edge or shell topology. This includes 2D curves and topology.

The following application objects are used by the wireframe_2d_model_with_topology UoF:

- Edge_based_2d_wireframe;
- Shell_based_2d_wireframe.

4.1.103 wireframe_with_topology

The wireframe_with_topology UoF defines the information needed to describe the representation of the shape of a product by wireframes that are defined by edge or shell topology. This includes 3D curves and topology.

The following application objects are used by the wireframe_with_topology UoF:

- Edge_based_wireframe;
- Shell_based_wireframe.

4.2 Application objects

This subclause specifies the application objects for the electronic assembly, interconnect, and packaging design application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

NOTE 1 This part of ISO 10303 describes the constraints on an application object in this clause in a form meant for domain users. Some of the constraints on an application objects are restated in clause 4.3 in a form meant for application developers.

NOTE 2 The syntax "application_object.name" represents the identifier for the attribute of the specified application object.

NOTE 3 The application objects may be related to other application objects by type relations, under constraints defined in the application object. If the relationships to other application objects exist, the application object should be treated a part of an instance network where the rest of the network constrains the application object being considered.

4.2.1 Absorbed_dose_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Absorbed_dose_data_element is a

The data associated with an Absorbed_dose_data_element are the following:

— additional_data.

4.2.1.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY absorbed_dose_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- absorbed_dose_data_element

( *
```

4.2.2 Activity_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Activity_data_element is a

The data associated with an Activity_data_element are the following:

— additional_data.

4.2.2.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY activity_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- activity_data_element

( *
```

4.2.3 Add_design_object_management_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Add_design_object_management_relationship is a type of Design_object_management_relationship (see 4.2.278).

NOTE Together with Change_design_object_management_relationship and Delete_design_object_management_relationship, Add_design_object_management_relationship supports the exchange of changes to a design by the use of a data set containing only (1) the information necessary to identify the design (e.g., an Ee_product_definition or an Ee_product_version), which is assumed to be already known to the receiving system, and (2) the design objects that have been added, changed, or deleted. Note that all the quality and structural requirements defined in this part of ISO 10303 may also apply to the resultant new design iteration.

The data associated with an Add_design_object_management_relationship are the following:

— current_design_object.

4.2.3.1 current_design_object

The current_design_object specifies the role of the Managed_design_object (see 4.2.524) for the Add_design_object_management_relationship.

Constraints:

UR1: The combination of the current_design inherited from the supertype Design_object_management_relationship and the current_design_object shall be unique within a population of Add_design_object_management_relationship.

NOTE EXPRESS specification:

```
*)
ENTITY add_design_object_management_relationship
  SUBTYPE OF (design_object_management_relationship);
  current_design_object : managed_design_object;
  UNIQUE
    UR1 : current_design_object, current_design;
END_ENTITY; -- add_design_object_management_relationship

(*
```

4.2.4 Adjacent_stratum_surface_embedded_component_surface_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Adjacent_stratum_surface_embedded_component_surface_definition is a type of Managed_design_object (see 4.2.524). An Adjacent_stratum_surface_embedded_component_surface_definition is a link in a sequence of stratum surfaces that result in the completed design. This link is not to another Stratum_surface but is instead to a surface of an Embedded_component. There may be multiple Adjacent_stratum_surface_embedded_component_surface_definition that reference the same stratum surface in the case where Stratum (see 4.2.845) are laterally adjacent. The use of Adjacent_stratum_surface_embedded_component_surface_definition in combination with the position in 2D space of the Embedded_component within the interconnect is sufficient to uniquely identify the 3D location of the Embedded_component.

The data associated with an Adjacent_stratum_surface_embedded_component_surface_definition are the following:

- name;
- precedent_item;
- subsequent_item.

4.2.4.1 name

The name specifies the string for the human-interpretable identifier for the Adjacent_stratum_surface_embedded_component_surface_definition.

4.2.4.2 precedent_item

The precedent_item specifies the previous Stratum_surface (see 4.2.868) or the Component_feature (see 4.2.118) for the Adjacent_stratum_surface_embedded_component_surface_definition.

4.2.4.3 subsequent_item

The subsequent_item specifies the previous Stratum_surface (see 4.2.868) or the Component_feature (see 4.2.118) for the Adjacent_stratum_surface_embedded_component_surface_definition.

Constraints:

UR1: The name shall be unique with a population of Adjacent_stratum_surface_embedded_component_surface_definition.

UR2: The combination of the precedent_item and the subsequent_item shall be unique within a population of Adjacent_stratum_surface_embedded_component_surface_definition.

WR1: The precedent_item shall be a secondary surface of a Stratum if the item is a Stratum_surface.

WR2: The subsequent_item shall be a primary surface of a Stratum if the item is a Stratum_surface.

NOTE EXPRESS specification:

```

*)
ENTITY adjacent_stratum_surface_embedded_component_surface_definition
SUBTYPE OF (managed_design_object);
    name : STRING;
    precedent_item : component_feature_or_stratum_surface;
    subsequent_item : component_feature_or_stratum_surface;
UNIQUE
    UR1 : name;
    UR2 : precedent_item, subsequent_item;
WHERE
    WR1: TYPEOF(precedent_item) &#60;&#62; TYPEOF(subsequent_item);
    WR2: NOT ('AP210_ARM.DATUM_FEATURE_USAGE_IN_DATUM_RELATIONSHIP' IN TYPEOF(SELF));
    WR3: precedent_item &#60;&#62; subsequent_item;
    WR4: NOT('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(precedent_item)) OR
        (SIZEOF(['AP210_ARM.BARE_DIE_SURFACE',
                'AP210_ARM.PACKAGE_BODY_SURFACE'] *
                TYPEOF(precedent_item.definition)) = 1);
    WR5: NOT('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(subsequent_item)) OR
        (SIZEOF(['AP210_ARM.BARE_DIE_SURFACE',
                'AP210_ARM.PACKAGE_BODY_SURFACE'] *
                TYPEOF(subsequent_item.definition)) = 1);
END_ENTITY; -- adjacent_stratum_surface_embedded_component_surface_definition

```

(*

4.2.5 Adjacent_stratum_surface_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Adjacent_stratum_surface_definition is a type of Managed_design_object (see 4.2.524). An Adjacent_stratum_surface_definition is a link in a sequence of stratum surfaces that result in the completed design. There may be multiple Adjacent_stratum_surface_definition that reference the same stratum surface in the case where Stratum (see 4.2.845) are laterally adjacent. The use of Adjacent_stratum_surface_definition in combination with the position in 2D space of the Stratum within the interconnect is sufficient to uniquely identify the 3D location of the Stratum.

The data associated with an Adjacent_stratum_surface_definition are the following:

- name;
- precedent_surface;
- subsequent_surface.

4.2.5.1 name

The name specifies the string for the human-interpretable identifier for the Adjacent_stratum_surface_definition.

4.2.5.2 precedent_surface

The precedent_surface specifies the previous Stratum_surface (see 4.2.868) for the Adjacent_stratum_surface_definition.

4.2.5.3 subsequent_surface

The subsequent_surface specifies the subsequent Stratum_surface (see 4.2.868) for the Adjacent_stratum_surface_definition.

Constraints:

UR1: The name shall be unique with a population of Adjacent_stratum_surface_definition.

UR2: The combination of the precedent_surface and the subsequent_surface shall be unique within a population of Adjacent_stratum_surface_definition.

WR1: The precedent_surface shall be a secondary surface of a Stratum.

WR2: The subsequent_surface shall be a primary surface of a Stratum.

NOTE EXPRESS specification:

```

*)
ENTITY adjacent_stratum_surface_definition
  SUBTYPE OF (managed_design_object);
    name          : STRING;
    precedent_surface : stratum_surface;
    subsequent_surface : stratum_surface;
  UNIQUE
    UR1 : name;
    UR2 : precedent_surface, subsequent_surface;
END_ENTITY; -- adjacent_stratum_surface_definition

( *
```

4.2.6 Advanced_b_rep

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Advanced_b_rep` is a type of `Bound_volume_shape` (see 4.2.76) that represents the shape of a part or an aspect of the shape of a part by an advanced boundary representation solid model. This representation allows for the curves and surfaces and the topology that binds them. Boundaries are explicitly defined only by topology. All of the geometry that defines the shape of a part or an aspect of the shape of a part must be associated with topology.

NOTE EXPRESS specification:

```
*)
ENTITY advanced_b_rep
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- advanced_b_rep

( *
```

4.2.7 Aggregate_connectivity_requirement

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Aggregate_connectivity_requirement` is a type of `Managed_design_object` (see 4.2.524). An `Aggregate_connectivity_requirement` is the association between a set of one or more `Functional_unit_network_node_definition` (see 4.2.403) and one `Design_composition_path` (see 4.2.262) that ensures that the requirement for connectivity is maintained between the functional and the physical designs.

The data associated with an `Aggregate_connectivity_requirement` are the following:

- `connectivity_requirement`;
- `design_definition_path`.

4.2.7.1 connectivity_requirement

The `connectivity_requirement` specifies the role of the `Functional_unit_network_node_definition` (see 4.2.403) for the `Aggregate_connectivity_requirement`.

4.2.7.2 design_definition_path

The `design_definition_path` specifies the role of the `Design_composition_path` (see 4.2.262) for the `Aggregate_connectivity_requirement`.

Constraints:

UR1: The `design_definition_path` shall be unique within a population of `Aggregate_connectivity_requirement`s.

WR1: Each of the terminals that participate in a connectivity_requirement shall be reachable from any other terminal in the design_definition_path that is specified by the same connectivity_requirement.

NOTE Since the hierarchical product structure is represented in this part of ISO 10303 as a directed acyclic graph, the closure requirement is evaluated by traversing the combined graph including both product structure and nodal representation. There is an individual nodal representation for each node or level of the product structure.

NOTE EXPRESS specification:

```

*)
ENTITY aggregate_connectivity_requirement
  SUBTYPE OF (managed_design_object);
    connectivity_requirement : functional_unit_network_node_definition;
    design_definition_path   : design_composition_path_or_root;
  UNIQUE
    UR1 : design_definition_path, connectivity_requirement;
  WHERE
    WR1: NOT ('AP210_ARM.FUNCTIONAL_UNIT_NETWORK_DEFINITION' IN
              TYPEOF(design_definition_path)) OR
          (connectivity_requirement.associated_functional_unit_definition =
           design_definition_path);
    WR2: NOT ('AP210_ARM.DESIGN_COMPOSITION_PATH' IN
              TYPEOF(design_definition_path)) OR
          (connectivity_requirement.associated_functional_unit_definition =
           design_definition_path.design_functional_unit.definition);
--tt tc1 wd12 modified
(*)
WHERE
  should not allocate this to both a physical_connectivity_definition
and to a physical_network for the same revision of the physical
product.
*)
END_ENTITY; -- aggregate_connectivity_requirement

(*)

```

4.2.8 Altered_package

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Altered_package is a type of Package (see 4.2.573). An Altered_package describes a Package that has been modified in some manner from a reference Package. The material or shape properties of the Altered_package are different from the reference Package.

EXAMPLE 1 Examples of alteration include trimming of the terminals, prebending of the terminals, tinning of the terminals for solderability, or more than one of these.

EXAMPLE 2 Figure 2 illustrates some of the different forms a package may take when it is installed into an assembly. The original form on the left is in the role of to_be_altered for the forms on the right. The altered forms are identified and tracked by an enterprise.

NOTE Most Altered_packages are a result of terminal alteration.

The data associated with an Altered_package are the following:

— package_to_be_altered.

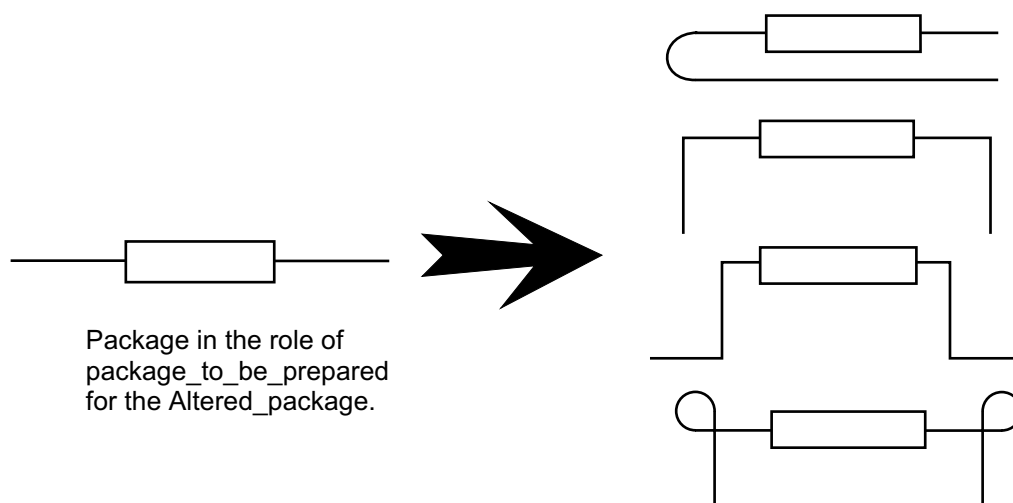


Figure 2 – Altered_package

4.2.8.1 package_to_be_altered

The package_to_be_altered specifies the Package (see 4.2.573) that is a reference for the Altered_package.

Constraints:

WR1: A Package shall not be in the role of package_to_be_altered for itself.

NOTE EXPRESS specification:

```

*)
ENTITY altered_package
  SUBTYPE OF (package);
    of_geometric_status : geometric_status;
    modified_terminal_separation : OPTIONAL length_data_element;
    package_to_be_altered : package;
END_ENTITY; -- altered_package
( *

```

4.2.9 Altered_package_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Altered_package_terminal is a type of Package_terminal (see 4.2.581). An Altered_package_terminal is one or more of a Length_trimmed_terminal (see 4.2.512), a Shape_formed_terminal (see 4.2.817), or a Surface_prepped_terminal (see 4.2.881). An Altered_package_terminal describes a Package_terminal that is the result of a modification process. Material or shape properties are different from the reference terminal.

EXAMPLE 1 The modification process may be performed prior to installing the occurrence of the Packaged_part (see 4.2.591) onto a Printed circuit board.

The data associated with an Altered_package_terminal are the following:

- associated_definition;
- terminal_to_be_altered.

4.2.9.1 associated_definition

The associated_definition specifies the role of the Altered_package (see 4.2.8) for the Altered_package_terminal.

4.2.9.2 terminal_to_be_altered

The terminal_to_be_altered specifies the Package_terminal (see 4.2.581) that will be modified for the Altered_package_terminal.

Constraints:

WR1: A Package_terminal shall not be in the role of terminal_to_be_altered for itself either directly or indirectly.

NOTE EXPRESS specification:

```

*)
ENTITY altered_package_terminal
  ABSTRACT SUPERTYPE OF ((shape_formed_terminal ANDOR
    surface_prepped_terminal) ANDOR length_trimmed_terminal)
  SUBTYPE OF (package_terminal);
    terminal_to_be_altered : package_terminal;
    SELF\part_feature.associated_definition : altered_package;
END_ENTITY; -- altered_package_terminal

( *

```

4.2.10 Altered_packaged_part

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Altered_packaged_part is a type of Packaged_part (see 4.2.591) whose used_package is an Altered_package (see 4.2.8). The material or shape properties are different from the reference Packaged_part.

NOTE For packaged parts that may have several alternate configurations (e.g., different lead configurations), an "unaltered" configuration can be designated using the base_packaged_part attribute. This base_packaged_part is then referenced by each Altered_packaged_part, along with the used_package, which references the Altered_package for that Altered_packaged_part. This relationship supports the traceability of altered parts, as well as the capability to readily identify potential replacement parts if a specific, altered part is out of stock, for example.

The data associated with an Altered_packaged_part are the following:

- base_packaged_part;
- used_package.

4.2.10.1 base_packaged_part

The base_packaged_part specifies the original, unmodified, Packaged_part (see 4.2.591) for the Altered_packaged_part.

4.2.10.2 used_package

The used_package specifies the Altered_package (see 4.2.8) that is the result of preparing the Package (see 4.2.573) by the enterprise for the Altered_packaged_part.

Constraints:

WR1: The Package in the role of used_package for the Packaged_part in the role of base_packaged_part shall be the same Package as the Package in the role of package_to_be_prepared for the Altered_package in the role of used_package.

WR2: A Packaged_part shall not be a base_packaged_part for itself, either directly or indirectly.

NOTE 1 This is a conceptual description only. It is understood that an enterprise does not disassemble a packaged part to accomplish its preparation. The view taken is that the package is the only item modified.

NOTE 2 It is recommended to use Engineering_make_from (see 4.2.353) For functional modifications (e.g., re-programming).

NOTE EXPRESS specification:

```

*)
ENTITY altered_packaged_part
  SUBTYPE OF (packaged_part);
    SELF\packaged_part.used_package : altered_package;
    base_packaged_part                : packaged_part;
END_ENTITY; -- altered_packaged_part

( *
```

4.2.11 Alternate_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Alternate_product is a type of Alternate_select_product (see 4.2.12) that is completely interchangeable with a specified Ee_product (see 4.2.319) with respect to form, fit, and function for an enterprise.

The data associated with an Alternate_product are the following:

— defined_alternate_product.

4.2.11.1 defined_alternate_product

The defined_alternate_product specifies the role of the Ee_product (see 4.2.319) for the Alternate_product.

NOTE EXPRESS specification:

```

*)
ENTITY alternate_product
  SUBTYPE OF (alternate_select_product);
  SELF\alternate_select_product.defined_alternate_product : SET [1:1] OF
  ee_product;
END_ENTITY; -- alternate_product

( *
```

4.2.12 Alternate_select_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Alternate_select_product is a type of Managed_design_object (see 4.2.524). An Alternate_select_product is one of an Alternate_product (see 4.2.11), an Assembly_alternate_product (see 4.2.32), or a Test_select_product (see 4.2.897). An Alternate_select_product specifies interchangeability relations at the product level. Form, fit and function evaluation by an enterprise is used to inform the decision to establish the relationship.

The data associated with an Alternate_select_product are the following:

- authorizing_agreement;
- defined_alternate_product;
- defining_base_product.

4.2.12.1 authorizing_agreement

The authorizing_agreement specifies the Contract (see 4.2.196) under which the Ee_product (see 4.2.319) was designed and procured for the Alternate_select_product. The authorizing_agreement need not be specified for a particular Alternate_select_product.

4.2.12.2 defined_alternate_product

The defined_alternate_product specifies one role of the Ee_product (see 4.2.319) for the Alternate_select_product. There shall be one or more Ee_product for the Alternate_select_product.

4.2.12.3 defining_base_product

The defining_base_product specifies one role of the Ee_product (see 4.2.319) for the Alternate_select_product. The defining_base_product is the reference with respect to form, fit, and function.

NOTE EXPRESS specification:

```

*)
ENTITY alternate_select_product
  ABSTRACT SUPERTYPE OF (ONEOF (assembly_alternate_product,
    test_select_product, alternate_product))
  SUBTYPE OF (managed_design_object);
  defined_alternate_product : SET [1:?] OF ee_product;
  defining_base_product    : ee_product;
  authorizing_agreement    : OPTIONAL contract;
END_ENTITY; -- alternate_select_product

```

(*

4.2.13 Amount_of_substance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Amount_of_substance_data_element is a

The data associated with an Amount_of_substance_data_element are the following:

— additional_data.

4.2.13.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY amount_of_substance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- amount_of_substance_data_element

```

(*

4.2.14 Analytical_model

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Analytical_model` is a type of `Managed_design_object` (see 4.2.524) that provides a mathematical description of the properties of a system.

NOTE 1 In this part of ISO 10303 an `Analytical_model` includes the variable declarations of the mathematical description but may not include the assignment of actual values for the variables declared. Assignment of actual values is accomplished in `Analytical_representation` (see 4.2.22)

NOTE 2 This part of ISO 10303 provides support for computer systems to verify type consistency between product data defined in this part of ISO 10303 and product data captured by `Analytical_models`.

NOTE 3 This part of ISO 10303 describes the interfaces (ports) to an `Analytical_model` and provides support for type checking of the units used for the parameters that may be assigned values for an `Analytical_model`.

EXAMPLE 1 Figure 3 is an example of the case where actual values are not included: the `Analytical_model` for a resistor that is coded in pseudocode.

When the `Analytical_model` is referenced by an `Analytical_representation` (see 4.2.22), literals will be supplied for items declared in the interface; both connections and their parameters, and the simulator must ensure that types are compatible.

NOTE 4 Usually the system is exercised in experiments to evaluate the usefulness of the system in the intended application.

NOTE 5 The language, syntax, and internal semantics of an `Analytical_model` are not specified by this part of ISO 10303.

NOTE 6 This part of ISO 10303 provides complete support for exchange of units, including SI units, derived units, and user declared units.

NOTE 7 This part of ISO 10303 provides complete support for prefixes of units.

The data associated with an `Analytical_model` are the following:

- `access_mechanism`;
- `name`;
- `parameter`;
- `reference_document`;
- `representation_language`;
- `source_code`.


```

Begin Interface declaration:
name := resistor;
ports{ } := (port 1, port 2);
parameter 1 name := resistance;
parameter 1 units := ohms;
End Interface declaration
-----
representation_language := "a simulation language definition document"
reference_document := some external definition

(* the following literals would be defined in the representation_language
current,
voltage,
units,
prefix,
:=
:=:
*
-
*

Ohms may be predefined or not, as required by the users of the language
In any case, it will be defined in the representation_language
or in the reference_document. *)

Model behavior declaration

I(input) := -I(output) {constraint based on conservation law}
V := V(input) - V(output) {constraint based on conservation law}
R := resistance {name mapping}
V := I(input) * R {statement of Ohm's Law, the constitutive equation for
this analytical_model}

```

Figure 3 – Analytical_model of a resistor

4.2.14.1 access_mechanism

The access_mechanism is an inverse relationship that specifies that the existence of the Analytical_model is dependent on the existence of the Analytical_model_port (see 4.2.17) that specifies the Analytical_model as its accessed_analytical_model. There shall be one or more Analytical_model_port for an Analytical_model.

4.2.14.2 name

The name specifies the string that is the identifier for the Analytical_model.

4.2.14.3 parameter

The parameter specifies the role of the Model_parameter (see 4.2.549) for the Analytical_model. There shall be one or more Model_parameter for a particular Analytical_model. Figure 3 illustrates the use of parameters.

4.2.14.4 reference_document

The reference_document specifies the role of the Ee_document (see 4.2.316) for the Analytical_model. The reference_document includes interface specifications for Analytical_models of interest to the enterprise.

4.2.14.5 representation_language

The representation_language specifies the Language_reference_manual (see 4.2.497) that defines the semantics and syntax of the computer interpretable strings in which the Analytical_model will be encoded. Figure 3 illustrates how the representation_language is specified and coded.

NOTE Only representation_languages that use characters from the ASCII code are supported by this part of ISO 10303.

4.2.14.6 source_code

The source_code specifies the role of the Ee_specification (see 4.2.338) for an Analytical_model. The source_code contains the source code for the Analytical_model. Figure3 illustrates how a source_code is specified and coded.

Constraints:

UR1: The name shall be unique within a population of Analytical_model.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model
  SUBTYPE OF (managed_design_object);
    name : ee_name;
    representation_language : language_reference_manual;
    reference_document : ee_document;
    parameter : SET [1:?] OF model_parameter;
    source_code : ee_specification;
    associated_definition : analytical_model_definition;
  INVERSE
    access_mechanism : SET [1:?] OF analytical_model_port FOR
      accessed_analytical_model;

  UNIQUE
    UR1 : name;
END_ENTITY; -- analytical_model

( *
```

4.2.15 Analytical_model_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_definition is a type of Ee_product_definition (see 4.2.323).

The data associated with an Analytical_model_definition are the following:

— associated_requirement_occurrence.

4.2.15.1 associated_requirement_occurrence

The associated_model_occurrence is an inverse relationship that specifies that the existence of the Analytical_model_definition is dependent on the existence of an Analytical_model to reference the Analytical_model_definition.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_definition
  SUBTYPE OF (ee_product_definition);
  INVERSE
    associated_model_occurrence :
      analytical_model FOR associated_definition;
END_ENTITY; -- analytical_model_definition

( *
```

4.2.16 Analytical_model_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_make_from_relationship is a type of Managed_design_object (see 4.2.524).

The data associated with an Analytical_model_make_from_relationship are the following:

- reusable_model;
- resultant_model.

4.2.16.1 reusable_model

The reusable_model specifies an Analytical_model (see 4.2.14) for the Analytical_model_make_from_relationship.

4.2.16.2 resultant_model

The resultant_model specifies an Analytical_model (see 4.2.14) for the Analytical_model_make_from_relationship.

Constraints:

UR1: The combination of reusable_model and resultant_model shall be unique within a population of Analytical_model_make_from_relationship.

WR1: The reusable_model shall not be the resultant_model within a population of Analytical_model_make_from_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_model : analytical_model;
    resultant_model : analytical_model;
UNIQUE
  UR1: reusable_model, resultant_model;
WHERE
  WR1: reusable_model:~::~ resultant_model;
END_ENTITY; -- analytical_model_make_from_relationship

( *
```

4.2.17 Analytical_model_port

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_port is a type of Managed_design_object (see 4.2.524). An Analytical_model_port may be an Analytical_model_vector_port (see 4.2.21), or additionally may be a Digital_analytical_model_port (see 4.2.290). An Analytical_model_port describes an interface point for an Analytical_model (see 4.2.14).

NOTE These points are the points at which a simulator allows a user to stimulate an instance of an Analytical_model or to evaluate the response of an instance of an Analytical_model to a stimulus in an experiment. In this context, instance includes the reference to the Analytical_model and the literals required for the parameters. A simulator that enforces conservative laws enforces that the energy entering an instance of an Analytical_model must be the same as that leaving the instance of the Analytical_model. The structure of the port is not visible merely by accessing this information. The subtypes of this object provide more visibility into the underlying structure as appropriate. Members of the subtype tree of this object are populated as needed by an application in order to disclose the appropriate level of detail. The specialization in the subtypes of this object occur as a result of being more specific about certain properties of the port.

The data associated with an Analytical_model_port are the following:

- accessed_analytical_model;
- distributed_properties;
- nominal_signal_flow_direction;
- port_name;
- port_type;
- unit_of_measure.

4.2.17.1 accessed_analytical_model

The accessed_analytical_model specifies the Analytical_model (see 4.2.14) that contains this Analytical_model_port.

NOTE 1 Containment in this case explicitly states that if a member of Analytical_model is deleted from the information base then the members of Analytical_model_port that are contained in that member of Analytical_model are also deleted.

NOTE 2 Containment in this case also explicitly states that the visibility of the name of an Analytical_model_port is limited to the Analytical_model that is the accessed_analytical_model.

4.2.17.2 distributed_properties

The distributed_properties specifies that the properties are either a distributed or a lumped element approximation for the Analytical_model_port. A value of TRUE means that the properties are distributed for the Analytical_model_port. A value of FALSE means that the properties are not distributed.

NOTE Multiple modes should be handled by assigning each mode a different port.

4.2.17.3 nominal_signal_flow_direction

The nominal_signal_flow_direction specifies either BIDIRECTIONAL, INPUT_DIRECTION, NOT_APPLICABLE, OUTPUT_DIRECTION, or UNKNOWN_DIRECTION for the Analytical_model_port. The nominal_signal_flow_direction need not be specified for a particular Analytical_model_port.

4.2.17.4 port_name

The port_name specifies the string that represents an identifier, understandable by the end user, for this Analytical_model_port.

NOTE The port_name is documented in the reference_document for the Analytical_model (see 4.2.14).

4.2.17.5 port_type

The port_type specifies one of a boolean_property_type, a logical_property_type, a physical_property_type, a numeric_property_type, or a string_property_type for this Analytical_model_port.

4.2.17.6 unit_of_measure

The unit_of_measure specifies the string for the descriptive label for the unit of measure of the value of a property associated with the simulated physical phenomenon exchanged through a member of the Analytical_model_port. The unit_of_measure need not be specified for a particular Analytical_model_port. The representation of the unit_of_measure is in accordance with ISO 10303-41.

Constraints:

UR1: The combination of the port_name and the accessed_analytical_model shall be unique within a population of Analytical_model_port.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_port
  SUPERTYPE OF (analytical_model_vector_port ANDOR digital_analytical_model_port)
  SUBTYPE OF (managed_design_object);
  accessed_analytical_model      : analytical_model;
  port_name                     : ee_name;
  unit_of_measure                : OPTIONAL STRING;
  port_type                     : model_parameter_type;
  distributed_properties         : BOOLEAN;
  nominal_signal_flow_direction : OPTIONAL signal_flow_direction;
  UNIQUE
  UR1 : port_name, accessed_analytical_model;
END_ENTITY; -- analytical_model_port

( *

```

4.2.18 Analytical_model_port_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_port_assignment is a type of Managed_design_object (see 4.2.524). An Analytical_model_port_assignment is the relationship between a Functional_unit_usage_view_terminal_definition (see 4.2.411) and an Analytical_model_port (see 4.2.17) for the purpose of conducting analyses of the design. An Analytical_model_port_assignment binds or associates a behavioural representation to a structural representation. The cardinality or size of the Analytical_model_port shall be consistent with the cardinality or size of the terminal in the structural representation.

NOTE This mechanism allows model ports and functional terminal definitions to be associated regardless of their individual names.

The data associated with an Analytical_model_port_assignment are the following:

- assigned_analytical_model_port;
- assigned_functional_unit_terminal.

4.2.18.1 assigned_analytical_model_port

The assigned_analytical_model_port specifies the role of the Analytical_model_port (see 4.2.17) for the Analytical_model_port_assignment.

4.2.18.2 assigned_functional_unit_terminal

The assigned_functional_unit_terminal specifies the role of the Functional_unit_usage_view_terminal_definition (see 4.2.411) for the Analytical_model_port_assignment.

NOTE Usually, only one Analytical_model is bound to a Functional_unit_definition at a time.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_port_assignment
  SUBTYPE OF (managed_design_object);
    assigned_functional_unit_terminal : functional_unit_usage_view_terminal_definition;
    assigned_analytical_model_port    : analytical_model_port;
WHERE
  WR1: NOT (('AP210_ARM.ANALYTICAL_MODEL_SCALAR_PORT' IN TYPEOF(assigned_analytical_model_po
    XOR ('AP210_ARM.SCALAR_TERMINAL_DEFINITION' IN TYPEOF(assigned_functional_unit_termi
END_ENTITY; -- analytical_model_port_assignment

( *

```

4.2.19 Analytical_model_port_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_port_make_from_relationship is a type of Managed_design_object (see 4.2.524).

The data associated with an Analytical_model_port_make_from_relationship are the following:

- associated_make_from;
- reusable_feature;
- resultant_feature.

4.2.19.1 associated_make_from

The associated_make_from specifies an Analytical_model_make_from_relationship (see 4.2.16) for the Analytical_model_port_make_from_relationship.

4.2.19.2 reusable_feature

The reusable_feature specifies an Analytical_model_port (see 4.2.17) for the Analytical_model_port_make_from_relationship.

4.2.19.3 resultant_feature

The resultant_feature specifies an Analytical_model_port (see 4.2.17) for the Analytical_model_port_make_from_relationship.

Constraints:

UR1: The combination of reusable_feature and resultant_feature shall be unique within a population of Analytical_model_port_make_from_relationship.

WR1: The reusable_feature shall not be the resultant_feature within a population of Analytical_model_port_make_from_relationship.

WR2: The Analytical_model referenced by the reusable_feature shall not be the Analytical_model referenced by the resultant_feature within a population of Analytical_model_port_make_from_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_port_make_from_relationship
  SUBTYPE OF (managed_design_object);
    associated_make_from : analytical_model_make_from_relationship;
    reusable_feature      : analytical_model_port;
    resultant_feature     : analytical_model_port;
UNIQUE
  UR1: reusable_feature, resultant_feature;
WHERE
  WR1: reusable_feature :&#60;&#62;; resultant_feature;
  WR2: reusable_feature.accessed_analytical_model :&#60;&#62;;
      resultant_feature.accessed_analytical_model;
END_ENTITY; -- analytical_model_port_make_from_relationship

( *
```

4.2.20 Analytical_model_scalar_port

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_scalar_port is a type of Analytical_model_vector_port (see 4.2.21) that has a size of exactly one. The Analytical_model_scalar_port is used if there is a requirement to specify the cardinality of the ports associated by a relationship entity.

EXAMPLE 1 Some examples are in the allocation to a functional unit terminal definition for analysis, or in mapping between a distributed model and a lumped equivalent circuit model, or in mapping to a vector port specified in an external syntax.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_scalar_port
  SUBTYPE OF (analytical_model_vector_port);
WHERE
  WR1: SELF\analytical_model_vector_port.size = 1;
END_ENTITY; -- analytical_model_scalar_port

( *

```

4.2.21 Analytical_model_vector_port

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_model_vector_port is a type of Analytical_model_port (see 4.2.17) that has a size greater than one. An Analytical_model_vector_port may be an Analytical_model_scalar_port (see 4.2.20). The Analytical_model_vector_port is used in the converse case to the Analytical_model_scalar_port. This part of ISO 10303 does not predefine the elements of a vector port.

NOTE The cardinality of the structured port is known and a list of names is specified for it. Examples are bus ports, multi-degrees-of-freedom connections, microwave domain modes, and address and data bus definitions.

The data associated with an Analytical_model_vector_port are the following:

— size.

4.2.21.1 size

The size specifies the integer greater than one for the cardinality of the Analytical_model_vector_port.

Constraints:

WR1: The size shall be greater than one.

NOTE EXPRESS specification:

```

*)
ENTITY analytical_model_vector_port
  SUPERTYPE OF (analytical_model_scalar_port)
  SUBTYPE OF (analytical_model_port);
  size : positive_integer;
END_ENTITY; -- analytical_model_vector_port

( *

```

4.2.22 Analytical_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Analytical_representation is a type of Managed_design_object (see 4.2.524). An Analytical_representation is the association of specific data to properties of an Analytical_model (see 4.2.14) in order to unambiguously characterize some behaviour of a real or imaginary object. The characterization may be considered to describe the behaviour exhibited by the model in a specific environment or for a range of environments for a specific experiment.

NOTE 1 This entity accomplishes a function similar to the parameter assignment part of a statement in a Spice netlist, or a function or subroutine call in a computer program. This capability is useful where the parts in the library have many parameters, not all of which apply to each simulation model that could be used for the part. This entity matches up the correct parameter values with the correct model.

NOTE 2 The properties specified should be in accordance with the capabilities and limitations of the Analytical_model. That is, the mathematical formulations in the Analytical_model apply over limited ranges of real product characteristics and environmental characteristics.

NOTE 3 This part of ISO 10303 does not standardize qualification of Analytical_representations for an intended usage.

The data associated with an Analytical_representation are the following:

- analytical_representation_name;
- model_parameter_assignment;
- model_representation.

4.2.22.1 analytical_representation_name

The analytical_representation_name specifies the string for the human-interpretable identifier for this Analytical_representation.

4.2.22.2 model_parameter_assignment

The model_parameter_assignment specifies the role of the Parameter_assignment (see 4.2.603) for the Analytical_representation. The model_parameter_assignment binds instance data to data slots in the Analytical_model. There shall be one or more Parameter_assignment for the Analytical_representation.

NOTE For each parameter declared in a model definition, an actual value must be assigned when that model is referenced, unless there is a default assignment included in the source code for the model.

4.2.22.3 model_representation

The model_representation specifies the Analytical_model (see 4.2.14) as the basis model for the Analytical_representation.

Constraints:

UR1: The analytical_representation_name shall be unique within a population of Analytical_representation.

WR1: Each member of model_parameter_assignment.assigned_parameter shall be a member of model_representation.model_parameters.

NOTE Only parameters declared in the model_representation are assigned values.

NOTE EXPRESS specification:

```
*)
ENTITY analytical_representation
  SUBTYPE OF (managed_design_object);
  model_parameter_assignment : SET [1:?] OF parameter_assignment;
  model_representation       : analytical_model;
  analytical_representation_name : ee_name;
  UNIQUE
  UR1 : analytical_representation_name;
END_ENTITY; -- analytical_representation

( *
```

4.2.23 Angle_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Angle_data_element is a type of Data_element (see 4.2.211) where the requisite unit type is that of angular displacement. An Angle_data_element may be combined with a Qualified_data_element (see ??) in a complex instance to include uncertainty in an angle.

NOTE EXPRESS specification:

```
*)
ENTITY angle_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- angle_data_element

( *
```

4.2.24 Angular_dimension_with_direction_vector

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Angular_dimension_with_direction_vector` is a type of `Dimensional_location` (see 4.2.300) that defines the relationship between two `Shape_element` (see 4.2.810) in terms of the angle between them and a vector that provides an orientation to that angle. The angle is measured in the plane normal to the specified direction vector and in a clockwise direction as viewed in the direction specified by the vector. The measurement is from the `Shape_element` specified as the `locating_shape_element` to the `Shape_element` specified as the `located_shape_element` by the characterized `Shape_element_locating_relationship` (see 4.2.815).

The data associated with an `Angular_dimension_with_direction_vector` are the following:

— `measurement_orientation`.

4.2.24.1 `measurement_orientation`

The `measurement_orientation` specifies either the `Datum_based_vector_orientation` (see 4.2.220) or an `Axis_placement` (see 4.2.56). If the `Datum_based_vector_orientation` is specified as the `measurement_orientation`, the specified `Datum_based_vector_orientation` provides the orientation by which the angle is measured relative to Datum (see 4.2.214). If an `Axis_placement` is specified, the orientation is specified in the context of the representation of the `Shape_elements` the relationship of which is characterized by the `Angular_dimension_with_direction_vector`.

Constraints:

UR1: The combination of `located_elements` inherited from `Dimensional_location` and `measurement_orientation` shall be unique within a population of `Angular_dimension_with_direction_vector`.

WR1: The `dimension_type` of the `Angular_dimension_with_direction_vector` shall be "angular".

NOTE The attribute `dimension_type` is declared in the `Dimensional_characteristic` entity of which this entity is a subtype.

WR2: The boolean value specified as `directed` shall be TRUE.

NOTE The attribute `directed` is declared in the `Dimensional_location` entity of which this entity is a subtype.

IP1: If the `measurement_orientation` specifies an `Axis_placement`, the `Axis_placement` shall be in the same context as the representations of the two `Shape_elements` that are located relative to each other by the `Angular_dimension_with_direction_vector`.

NOTE EXPRESS specification:

```

*)
ENTITY angular_dimension_with_direction_vector
  SUBTYPE OF (dimensional_location);
    measurement_orientation :
axis_placement_or_datum_based_vector_orientation_select;
  UNIQUE
    UR1 : SELF\dimensional_location.located_elements, measurement_orientation;
END_ENTITY; -- angular_dimension_with_direction_vector

( *

```

4.2.25 Angular_size_dimension

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Angular_size_dimension is a type of Dimensional_size (see 4.2.301) that characterizes the angularity of a Shape_element (see 4.2.810). If the Angular_size_dimension is a Single_boundary_dimension (see 4.2.827), the Angular_size_dimension specifies the angle formed by the characterized Shape_element. If the Angular_size_dimension is a Dimensional_size_based_on_opposing_boundaries (see 4.2.302), the Angular_size_dimension specifies the angle formed by two Shape_elements that make up the characterized Shape_element.

The data associated with an Angular_size_dimension are the following:

- half_angle;
- major_angle.

4.2.25.1 half_angle

The half_angle specifies the boolean value that indicates whether the full angle is being characterized by the Angular_size_dimension, or whether the angle that is being characterized is the angle between the characterized Shape_element (see 4.2.810) and a centre line or centre plane of the characterized Shape_element. A value of TRUE indicates that the angle between the Shape_element and a centre line or centre plane is being characterized. A value of FALSE indicates that the full angle is being characterized.

4.2.25.2 major_angle

The major_angle specifies the angle to which the Angular_size_dimension is applied. A value of LARGE indicates that it is the larger of the angles formed by the characterized Shape_element (see 4.2.810). A value of SMALL indicates that it is the smaller of the angles formed by the characterized Shape_element. A value of EQUAL indicates that it may be either of the angles formed by the characterized Shape_element, since they are defined to be equal.

Constraints:

WR1: The dimension_type shall be "angular".

NOTE The attribute dimension_type is declared in the Dimensional_characteristic entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY angular_size_dimension
  SUBTYPE OF (dimensional_size);
    half_angle : BOOLEAN;
    major_angle : angle_relator;
END_ENTITY; -- angular_size_dimension

( *
```

4.2.26 Angularity_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Angularity_tolerance is a type of Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that specifies or partially specifies the requirement on the angularity of a Shape_element (see 4.2.810) in relation to one or two Datum (see 4.2.214).

NOTE Angularity_tolerance is defined in terms of tolerance zone definitions in 14.9 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Centre_axis, Centre_plane, Physical_feature_or_part_template, or Tangent_plane.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: There shall be at most two Datum in the Datum_system, the definition of which is specified as the datum_system_definition.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

WR3: At least one Datum_plane or Datum_axis shall be in the Datum_system, the definition of which is specified as the datum_system_definition.

WR4: If the `toleranced_shape_element` is a `Centre_plane`, `Physical_feature_or_part_template`, or `Tangent_plane`, the `Angularity_tolerance` shall specify a `Tolerance_zone_definition` as its `tolerance_zone_or_boundary_definition`; and that `Tolerance_zone_definition` shall specify a `Tolerance_zone_boundary_definition_with_specified_size` as its `boundary_definition`; and that `Tolerance_zone_boundary_definition_with_specified_size` shall specify an `Opposing_boundary_set_size_characteristic` as its `specified_boundary_size`.

IP1: A `Parallelism_tolerance` as opposed to an `Angularity_tolerance` shall be applied if the requirement is on the angularity of a `Shape_element` in relation to a `Datum_axis` or `Datum_plane` and the theoretically exact value of the angle is 0 or is an integral multiple of 180 degrees (π radians).

IP2: A `Perpendicularity_tolerance` as opposed to an `Angularity_tolerance` shall be applied if the requirement is on the angularity of a `Shape_element` in relation to a `Datum_axis` or `Datum_plane` and the theoretically exact value of the angle is an odd multiple of 90 degrees ($\pi/2$ radians).

IP3: The form of the `toleranced_shape_element` shall be consistent with the form of a shape element for which the application of an `angularity_tolerance` is sensible.

NOTE EXPRESS specification:

```
* )
ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- angularity_tolerance

( *
```

4.2.27 Apex

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Apex` is a type of `Derived_shape_element` (see 4.2.260) that is either a point that is derived by projecting a conical surface or the common point that is derived by projecting two or more conical surfaces. The `Shape_element` (see 4.2.810) from which the `Apex` is derived are associated with the `Apex` by the `Shape_element_deriving_relationship` (see 4.2.814) that are specified as the `deriving_relationships`.

NOTE The attribute `deriving_relationships` is declared in the `Derived_shape_element` entity of which this entity is a subtype.

Constraints:

IP1: The common intersection of three or more planes or two curves shall be treated as a `Geometric_intersection` rather than as an `Apex`.

IP2: The form of the `Shape_elements` from which the `Apex` is derived shall be consistent with conical surfaces from which an `Apex` can be derived.

NOTE EXPRESS specification:

```
*)
ENTITY apex
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- apex

--tt tc1wd2
(*
```

4.2.28 Area_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Area_data_element is a

The data associated with an Area_data_element are the following:

— additional_data.

4.2.28.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY area_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- area_data_element

(*
```

4.2.29 Array_placement_group

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Array_placement_group is a type of Assembly_group_component (see 4.2.40) that is a group of Assembly_component (see 4.2.34) placed in an linear or rectangular regular pattern in the assembly. An Array_placement_group may be a Laminate_text_string_component (see 4.2.489). The position tolerance specification of an Assembly_component that is included in an array is different from an Assembly_component that is randomly placed. Text support in this part of ISO 10303 is limited to a linear pattern defined by the character to character spacing defined in the font definition. Spacing properties for items other than text is supported by applying Geometric Dimensioning and Tolerancing methods.

NOTE This part of ISO 10303 provides support for multi-line messages with one instance of the Laminate_text_string_component for each line of the multi-line message.

EXAMPLE 1 Examples of components placed as arrays include: test points, thermal vias, integrated circuit memory devices, light emitting diodes, spare land areas, component terminal mating features where the component itself has terminals arranged in a grid, and text.

EXAMPLE 2 Figure 4 is an example layout for a component array. The Axis_placement (see 4.2.56) that is in the role of origin for the Package (see 4.2.573) definition in this example is placed at the centroid of the Package and consequently at the centroid of the Package_body (see 4.2.574). The illustration is typical of a drawing presentation of the design information conveyed by this part of ISO 10303. Each rectangle in the illustration is a representation of an instance of a Package_body.

NOTE EXPRESS specification:

```
*)
ENTITY array_placement_group
--tt tc1wd2
--tt tc1wd8 change
    ABSTRACT SUPERTYPE OF (ONEOF(laminate_text_string_component,
                                   linear_array_placement_group_component,
                                   rectangular_array_placement_group_component))
    SUBTYPE OF (assembly_group_component);
END_ENTITY; -- array_placement_group
(*)
```

4.2.30 Assembled_with_bonding

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembled_with_bonding is a type of Assembly_joint (see 4.2.41) that establishes the joint by bonding. Bonding requires use of material that may change state during the bonding process to form a permanent or semi-permanent joint. In this context, a permanent or semi-permanent joint means that repair of the affected part of the assembly would require destruction and replacement of the bond material.

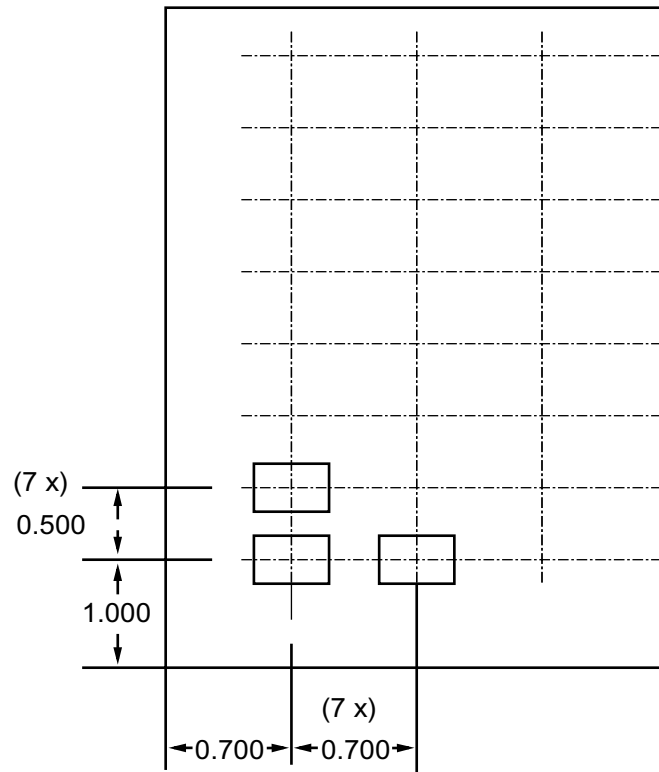


Figure 4 – Array_placement_group

NOTE The `Assembled_with_bonding` object relates two assembly features that are bonded and the `Assembly_bond_definition` that specifies the category of the bond, its shape, its material, and its name.

EXAMPLE 1 Figure 5 illustrates the concept of the joint category described herein and the roles of the related application objects.

The data associated with an `Assembled_with_bonding` are the following:

- `bond_assembly_position`;
- `default_bond_definition`.

4.2.30.1 bond_assembly_position

The `bond_assembly_position` is an inverse relationship that specifies that the `Assembled_with_bonding` shall be referenced by no more than one `Bond_assembly_position` as its bond. The `bond_assembly_position` need not be specified for a particular `Assembled_with_bonding`. The position is provided to identify shape placement for analysis.

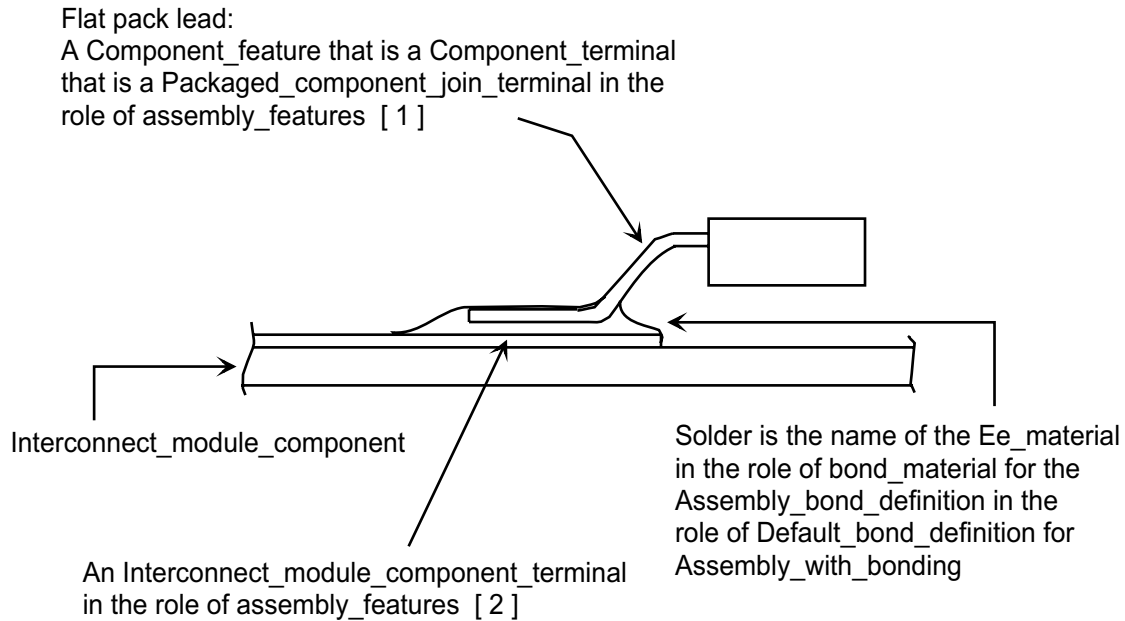


Figure 5 – Assembled_with_bonding

4.2.30.2 default_bond_definition

The default_bond_definition specifies the Assembly_bond_definition (see 4.2.33) used by the Assembled_with_bonding process to form a join.

EXAMPLE 1 A type of bond material is solder. A type of component feature is a component terminal.

Constraints:

WR1: The members of Part_feature utilized by Assembled_with_bonding that are inherited from Assembly_joint shall be the same members of Part_feature specified inclusively by Assembly_bond_definition bonded_feature_1 and bonded_feature_2.

NOTE EXPRESS specification:

```

*)
ENTITY assembled_with_bonding
  SUBTYPE OF (assembly_joint);
    default_bond_definition : assembly_bond_definition;
  INVERSE
    bond_assembly_position : SET [0:1] OF bond_assembly_position FOR bond;
--tt tc1 wd8 added
WHERE
  WR1: SELF\assembly_joint.assembly_features_1.definition :=:
                                default_bond_definition.bonded_feature_1;
  WR2: SELF\assembly_joint.assembly_features_2.definition :=:
                                default_bond_definition.bonded_feature_2;
END_ENTITY; -- assembled_with_bonding

( *

```

4.2.31 Assembled_with_fasteners

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembled_with_fasteners is a type of Assembly_joint (see 4.2.41) that establishes the assembly by using fasteners to mechanically position a component so that its component mounting features are connected to component mounting features of another component in the assembly. The fasteners are associated with the base component using Product_association (see 4.2.741).

EXAMPLE 1 The usual case for this is the mounting hardware associated with a power device mounted to the Interconnect_module_component (see 4.2.442). Figure 6 illustrates the concept of the joint category described herein and the roles of the related Application Objects.

The data associated with an Assembled_with_fasteners are the following:

— associated_assembly.

4.2.31.1 associated_assembly

The associated_assembly specifies the Assembly_group_component (see 4.2.40) that identifies the components required to be considered together in the assembly process for the Assembled_with_fasteners. The associated_assembly does not describe a formal product structure hierarchy.

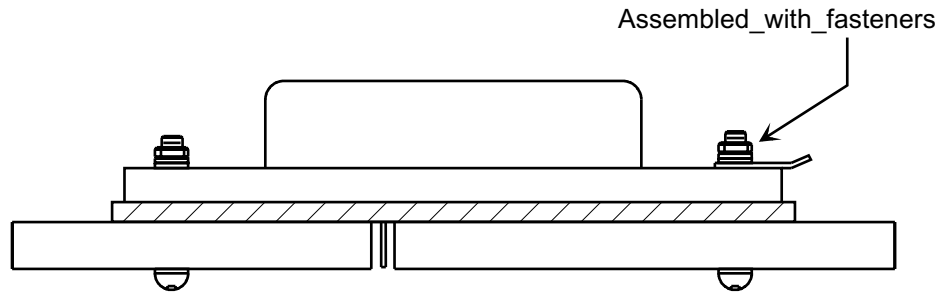


Figure 6 – Assembled_with_fasteners

NOTE EXPRESS specification:

```

*)
ENTITY assembled_with_fasteners
  SUBTYPE OF (assembly_joint);
  SELF\assembly_joint.associated_assembly : assembly_group_component;
END_ENTITY; -- assembled_with_fasteners

```

(*

4.2.32 Assembly_alternate_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An *Assembly_alternate_product* is a type of *Alternate_select_product* (see 4.2.12) where the alternate is valid only for a particular design. The alternate is valid for all instances of the *defining_base_product* inherited from *Alternate_select_product* within that design.

NOTE In specifying the components to be used in an assembly, it is a fairly common practice to stipulate that any usage of a given component (as referenced by a part number, for example) in that assembly may be replaced by some other specified component. Each of those other components is then an *Assembly_alternate_product* for the specified component (*defining_base_product*) for that assembly. The *Assembly_alternate_product* supports only this assembly-level alternate: i.e., it only refers to components that can serve as alternates for any usage of the specified component (the *defining_base_product*) in that assembly.

The data associated with an *Assembly_alternate_product* are the following:

— *basis_context*.

4.2.32.1 basis_context

The *basis_context* specifies the role of the *Physical_unit_design_view* (see 4.2.667) for the *Assembly_alternate_product*.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_alternate_product
  SUBTYPE OF (alternate_select_product);
    basis_context : physical_unit_design_view;
END_ENTITY; -- assembly_alternate_product

( *
```

4.2.33 Assembly_bond_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_bond_definition is a type of Managed_design_object (see 4.2.524). An Assembly_bond_definition is the definition of the joint that uses bulk material to bond together portions of the assembly.

NOTE This part of ISO 10303 provides the ability to define the typical shape of the bond through the use of Default_assembly_bond_shape (see 4.2.244).

The data associated with an Assembly_bond_definition are the following:

- analytical_assembly_bond_representation;
- associated_definition;
- bond_category;
- bond_material;
- bonded_feature_1;
- bonded_feature_2;
- default_shape;
- name.

4.2.33.1 analytical_assembly_bond_representation

The analytical_assembly_bond_representation specifies the role of Analytical_representation for the Assembly_bond_definition. There may be one or more analytical_assembly_bond_representation for Assembly_bond_definition. The analytical_assembly_bond_representation need not be specified for a particular Assembly_bond_definition.

4.2.33.2 associated_definition

The associated_definition specifies the role of the Externally_defined_product_definition (see ??) for the Assembly_bond_definition. The associated_definition is only provided to support identification of the product classification.

4.2.33.3 bond_category

The bond_category specifies the role of the External_definition (see 4.2.361) for the Assembly_bond_definition.

NOTE This part of ISO 10303 does not predefine the classification of bonds for assembly.

EXAMPLE 1 Some typical types are wire bond, solder, and conductive adhesive.

4.2.33.4 bond_material

The bond_material specifies the role of the Ee_material (see 4.2.318) for the Assembly_bond_definition. The bond material may be a composition.

4.2.33.5 bonded_feature_1

The bonded_feature_1 specifies one role of Part_feature for Assembly_bond_definition. The bonded_feature_1 defines one of the Part_feature that will be utilized by Assembled_with_bonding.

4.2.33.6 bonded_feature_2

The bonded_feature_2 specifies one role of Part_feature for Assembly_bond_definition. The bonded_feature_2 defines the second Part_feature that will be utilized by Assembled_with_bonding.

4.2.33.7 default_shape

The default_shape is an inverse relationship that specifies that there shall be no more than one Default_assembly_bond_shape (see 4.2.244) that specifies the Assembly_bond_definition as its characterized_bond_definition. The default_shape need not be specified for a particular Assembly_bond_definition.

4.2.33.8 name

The name specifies the string identifying the Assembly_bond_definition.

Constraints:

UR1: The name shall be unique within a population of Assembly_bond_definitions.

UR2: The item that is bonded_feature_1 shall not be bonded_feature_2.

WR1: There shall be a product category of 'technology specific model' assigned to the product this assembly_bond_definition is associated with.

WR2: The associated_definition shall not be another subtype of Ee_product_definition.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_bond_definition
  SUBTYPE OF (managed_design_object);
  analytical_assembly_bond_representation :
OPTIONAL SET [1:?] OF analytical_representation;
  associated_definition                    : externally_defined_ee_product_definition;
  bond_category                          : external_definition;
  bond_material                          : ee_material;
  bonded_feature_1                       : part_feature;
  bonded_feature_2                       : part_feature;
  name                                   : ee_name;
INVERSE
  default_shape : SET [0:1] OF default_assembly_bond_shape FOR
                      characterized_bond_definition;
UNIQUE
  UR1 : name;
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*
    USEDIN(associated_definition\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') |
      (prpc.name = 'technology specific model')
    )) &#62;= 0;
END_ENTITY; -- assembly_bond_definition

( *
```

4.2.34 Assembly_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_component is a type of Managed_design_object (see 4.2.524). An Assembly_component may be one of an Assembly_module_component (see 4.2.44), a Bare_die_component (see 4.2.61), an Interconnect_module_component (see 4.2.442), a Laminate_component (see 4.2.483), a Packaged_component (see 4.2.585), or a Printed_component (see 4.2.721), or additionally may be an Assembly_group_component (see 4.2.40), or additionally may be a Shield (see 4.2.821), or additionally may be a Thermal_component (see 4.2.900), or additionally may be a Routed_interconnect_component (see 4.2.777), or additionally may be an Interface_component (see 4.2.468). An Assembly_component is the occurrence of a Physical_unit (see 4.2.664) or Part_template (see 4.2.620) in an assembly. An Assembly_component may include additional product data over that defined in Physical_unit or Part_template. An Assembly_component that is no other type is intended to be an instance of a Physical_unit and not of a Part_template.

EXAMPLE 1 Figure 7 illustrates a detail of an assembly drawing that shows the Assembly_component in relation to a cross-sectional view of the Interconnect_module_component. Figure ?? illustrates a complete assembly showing multiple Assembly_component.

NOTE 1 An Assembly_component may be assembled onto an Interconnect_module_component in the realization of an Assembly_module (see 4.2.43). An Assembly_component may be an occurrence of an Assembly_module_usage_view (see 4.2.51). An Assembly_component may be an occurrence of an Interconnect_module_usage_view (see 4.2.465). For the purposes of this model, the current interpretation is to treat an embedded component as a Packaged_component. The Interconnect_module_component_terminal (see 4.2.445) that mates with the Packaged_component_join_terminal (see 4.2.586) of the embedded component is not visible when the Assembly_module is assembled, but the Interconnect_module_component_terminal is still logically the interface to the Interconnect_module_component for the embedded component.

The manufacturing view supports the fact that the embedded component is installed during part of the sequential manufacturing process for the Interconnect_module (see 4.2.440).

NOTE 2 An Assembly_component may be a floor-planning symbol if the definition is an Assembly_module or an Interconnect_module. If the Assembly_component is used in this manner, then the actual components will be inclusion_definitioned in the design in a later iteration that results from inclusion_definitioning the detailed components from the Assembly_module or Interconnect_module using a computer processor. The processor may choose to support traceability by using the Component_make_from_relationship (see 4.2.126) to specify what component of the Assembly_module or Interconnect_module corresponds to the component in the design.

NOTE 3 Parameters may be assigned to Assembly_component by the Planned_characteristic object through the Assembly_requirements_allocation.

The data associated with an Assembly_component are the following:

- assembly;
- component_2d_shape;
- component_3d_shape;
- usage_definition.

4.2.34.1 assembly

The assembly is an inverse relationship that specifies that the existence of the Assembly_component is dependent on the existence of the Assembly_composition_relationship (see 4.2.39) that specifies the Assembly_component as its component. There shall be one or more Assembly_composition_relationship for an Assembly_component.

NOTE This cardinality supports multiple versions of the design, with the same member of Assembly_component.

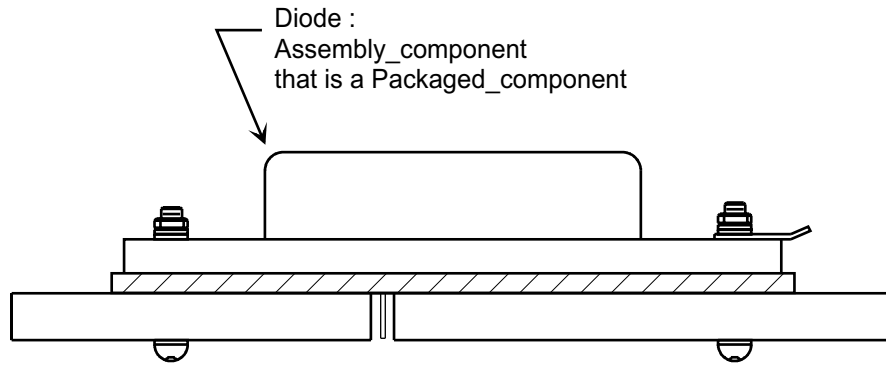


Figure 7 – Assembly_component

4.2.34.2 component_2d_shape

The component_2d_shape is an inverse relationship that specifies that there shall be no more than one Assembly_component_2d_shape (see 4.2.35) that specifies the Assembly_component as its shape_characterized_component. The component_2d_shape need not be supplied for a particular Assembly_component.

4.2.34.3 component_3d_shape

The component_3d_shape is an inverse relationship that specifies that there shall be no more than one Assembly_component_3d_shape (see 4.2.36) that specifies the Assembly_component as its shape_characterized_component. The component_3d_shape need not be supplied for a particular Assembly_component.

4.2.34.4 usage_definition

The usage_definition specifies either a Part_template (see 4.2.620) or a Physical_unit (see 4.2.664) for the Assembly_component. The usage_definition need not be supplied for an Assembly_component. The usage_definition is optional because some types of Assembly_component are defined through another mechanism. The types of Assembly_component for which usage_definition is required include the explicit statement of that requirement in their definition.

Constraints:

WR1: If the usage_definition exists, then only a Footprint_occurrence shall reference a Footprint_definition and only a Padstack_occurrence shall reference a Padstack_definition.

WR2: All members of assembly shall be for the same product.

WR3: No more than one member of assembly shall be for a version of the product.

WR4: An Assembly_component shall not be both a Laminate_component and an Interface_component.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_component
  ABSTRACT SUPERTYPE OF (((ONEOF (
    physical_component,
    printed_component,
    laminate_component)
    ANDOR shield)
    ANDOR thermal_component)
    ANDOR assembly_group_component)
  SUBTYPE OF (managed_design_object);
  usage_definition : OPTIONAL physical_unit_or_part_template_or_fp_or_pd;
  INVERSE
    assembly          : SET [1:?] OF assembly_composition_relationship FOR
                        component;
    component_2d_shape : SET [0:1] OF assembly_component_2d_shape FOR
                        shape_characterized_component;
    component_3d_shape : SET [0:1] OF assembly_component_3d_shape FOR
                        shape_characterized_component;
    component_non_planar_shape : SET [0:1] OF assembly_component_non_planar_shape FOR
                        shape_characterized_component;
  WHERE
    WR1: NOT(EXISTS(usage_definition)) OR
    (
      (('AP210_ARM.FOOTPRINT_OCCURRENCE' IN TYPEOF(SELF)) AND
      ('AP210_ARM.FOOTPRINT_DEFINITION' IN TYPEOF(usage_definition))) OR
      (NOT ('AP210_ARM.FOOTPRINT_OCCURRENCE' IN TYPEOF(SELF))
      AND NOT ('AP210_ARM.FOOTPRINT_DEFINITION' IN TYPEOF(usage_definition))) OR
      (('AP210_ARM.PADSTACK_OCCURRENCE' IN TYPEOF(SELF)) AND
      ('AP210_ARM.PADSTACK_DEFINITION' IN TYPEOF(usage_definition))) OR
      (NOT ('AP210_ARM.PADSTACK_OCCURRENCE' IN TYPEOF(SELF)) AND
      NOT ('AP210_ARM.PADSTACK_DEFINITION' IN TYPEOF(usage_definition)))
    );
END_ENTITY; -- assembly_component
(
  *

```

4.2.35 Assembly_component_2d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_component_2d_shape is a type of Planar_projected_shape (see 4.2.677). An Assembly_component_2d_shape is the two-dimensional shape of an Assembly_component (see 4.2.34).

NOTE The shape is defined by the `Physical_unit_planar_shape` (see 4.2.670), or by the equivalent for `Part_template` (see 4.2.620), referenced by the location transformation in `Component_assembly_2d_position` (see 4.2.112).

The data associated with an `Assembly_component_2d_shape` are the following:

— `shape_characterized_component`.

4.2.35.1 `shape_characterized_component`

The `shape_characterized_component` specifies the role of the `Assembly_component` (see 4.2.34) for the `Assembly_component_2d_shape`.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_component_2d_shape
  SUBTYPE OF (planar_projected_shape);
  shape_characterized_component : SET [1:?] OF assembly_component;
END_ENTITY; -- assembly_component_2d_shape

(*
```

4.2.36 `Assembly_component_3d_shape`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Assembly_component_3d_shape` is a type of `Feature_shape_definition` (see 4.2.372). An `Assembly_component_3d_shape` is the three-dimensional shape of an `Assembly_component` (see 4.2.34).

The data associated with an `Assembly_component_3d_shape` are the following:

— `shape_characterized_component`.

4.2.36.1 `shape_characterized_component`

The `shape_characterized_component` specifies the role of the `Assembly_component` (see 4.2.34) for the `Assembly_component_3d_shape`.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_component_3d_shape
  SUBTYPE OF (feature_shape_definition);
  shape_characterized_component : SET [1:?] OF assembly_component;
END_ENTITY; -- assembly_component_3d_shape

( *
```

4.2.37 Assembly_component_non_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_component_non_planar_shape is a type of Manifold_subsurface_shape (see 4.2.525), and a Feature_shape_definition (see 4.2.372). An Assembly_component_non_planar_shape is the shape of a Assembly_component (see 4.2.34) that is represented in the form of a surface.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_component_non_planar_shape
  SUBTYPE OF (manifold_subsurface_shape, feature_shape_definition);
  shape_characterized_component : SET [1:?] OF assembly_component;
END_ENTITY; -- assembly_component_non_planar_shape

( *
```

4.2.38 Assembly_component_placement_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_component_placement_link is a type of Managed_design_object (see 4.2.524).

The data associated with an Assembly_component_placement_link are the following:

- precedent_element;
- referenced_by;
- subsequent_element.

4.2.38.1 precedent_element

The precedent_element specifies an Assembly_component (see 4.2.34) for the Assembly_component_placement_link.

4.2.38.2 referenced_by

The `referenced_by` is an inverse relationship that specifies that the existence of the `Assembly_component_placement_link` is dependent on the existence of the `Linear_array_placement_group_component` (see 4.2.516) that specifies the `Assembly_component_placement_link` as its element.

4.2.38.3 subsequent_element

The `subsequent_element` specifies an `Assembly_component` (see 4.2.34) for the `Assembly_component_placement_link`.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_component_placement_link
  SUBTYPE OF (managed_design_object);
    precedent_element : assembly_component;
    subsequent_element : assembly_component;
INVERSE
  referenced_by : linear_array_placement_group_component FOR
                  element;
END_ENTITY; -- assembly_component_placement_link

( *
```

4.2.39 Assembly_composition_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Assembly_composition_relationship` is a type of `Managed_design_object` (see 4.2.524). An `Assembly_composition_relationship` may be a `Next_higher_assembly_relationship` (see 4.2.556). An `Assembly_composition_relationship` relates an `Assembly_component` (see 4.2.34) to the assembly that it helps to compose.

The data associated with an `Assembly_composition_relationship` are the following:

- `assembly`;
- `assembly_bonding_agent`;
- `component`;
- `security_code`.

4.2.39.1 assembly

The assembly specifies the Physical_unit_design_view (see 4.2.667) for the product realized by this Assembly_composition_relationship.

4.2.39.2 assembly_bonding_agent

The assembly_bonding_agent specifies the role of the Ee_material (see 4.2.318) for the Assembly_composition_relationship. There may be more than one Ee_material for the Assembly_composition_relationship. The assembly_bonding_agent need not be specified for a particular Assembly_composition_relationship.

NOTE 1 The use of this attribute is appropriate where no feature shape data is available.

NOTE 2 To identify quantity of bulk material, this part of ISO 10303 predefines Assembly_material_composition_relationship (see 4.2.42).

4.2.39.3 component

The component specifies the role of the Assembly_component (see 4.2.34) for the Assembly_composition_relationship.

4.2.39.4 security_code

The security_code specifies the role of the Security_classification (see 4.2.804) for the Assembly_composition_relationship. The security_code is optional and need not be supplied.

Constraints:

UR1: The combination of the assembly and the component shall be unique within a population of Assembly_composition_relationship.

WR1: At least one member of Assembly_composition_relationship shall specify a member of Assembly_component that is also a member of Interface_component.

NOTE This part of ISO 10303 requires the interface to the next level of assembly to exist. This is because this part of ISO 10303 supports design and archival of designs that may be used in higher levels of assembly.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_composition_relationship
  SUPERTYPE OF (ONEOF (next_higher_assembly_relationship))
  SUBTYPE OF (managed_design_object);
  assembly          : physical_unit_design_view;
  component          : assembly_component;
  security_code      : OPTIONAL security_classification;
  assembly_bonding_agent : OPTIONAL SET [1:?] OF ee_material;
  UNIQUE
    UR1 : assembly, component;
END_ENTITY; -- assembly_composition_relationship

( *

```

4.2.40 Assembly_group_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_group_component is a type of Assembly_component (see 4.2.34). An Assembly_group_component may be either one or both of an Array_placement_group (see 4.2.29) or an Interfaced_group_component (see 4.2.473). Each component that is part of a group is in the same design view as that group.

EXAMPLE 1 This application object is provided to support, for instance, the hardware items required for power device mounting.

NOTE 1 This application object is not included to support process plans.

NOTE 2 The intent is for all members of Assembly_component that are also in the role of related_components for Sub_assembly_relationship to be placed as a group in a placement activity using a CAD tool. However, each member of the group is placed individually in this standard. A receiving system should treat the relative placement of members of the group as derived data.

EXAMPLE 2 This application object supports paired-line layout by making each individual conductive_interconnect_element associated by the sub_assembly_relationship entity to this application object one of the lines to be paired. Extensions to higher number of lines is obvious.

The data associated with an Assembly_group_component are the following:

— components.

4.2.40.1 components

The components is an inverse relationship that specifies that the existence of the Assembly_group_component is dependent on the existence of the Sub_assembly_relationship (see 4.2.877) that specifies the Assembly_group_component as its relating_component. There shall be one or more Sub_assembly_relationship for an Assembly_group_component.

NOTE The usual case is for one member of Sub_assembly_relationship to associate all relating components, but more than one is permitted. There is no semantic difference in this part of ISO 10303 between all the components being related with one member of Sub_assembly_relationship or each of the component being related having its own Sub_assembly_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_group_component
  SUPERTYPE OF (interfaced_group_component ANDOR array_placement_group ANDOR
    (ONEOF(footprint_occurrence, padstack_occurrence)))
  SUBTYPE OF (assembly_component);
  INVERSE
    components : SET [1:?] OF sub_assembly_relationship FOR relating_component;
END_ENTITY; -- assembly_group_component
( *
```

4.2.41 Assembly_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_joint is a type of Managed_design_object (see 4.2.524). An Assembly_joint may be either one of or both of an Assembled_with_bonding (see 4.2.30) or an Assembled_with_fasteners (see 4.2.31) or additionally may be a Connection_zone_based_assembly_joint (see 4.2.186). An Assembly_joint is the result of the process that establishes physical connectivity between component features of two components in an assembly.

EXAMPLE 1 Figure 8 illustrates the roles assigned to items in an assembly by a designer in constructing a member of Assembly_joint.

NOTE 1 Assembly_joint is a central concept in this part of ISO 10303 that provides information necessary to extract a network description of a design from the assembly model. Once extracted, the description may be compared to the original requirements.

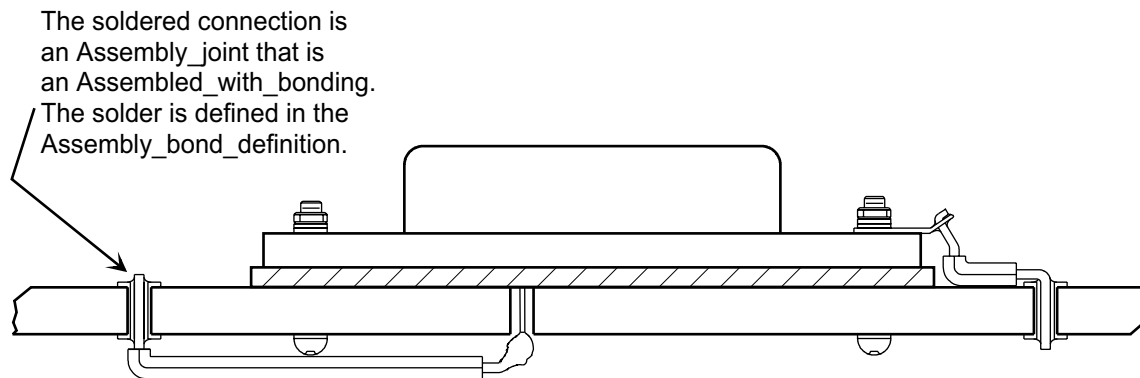
NOTE 2 Assembly_joint provides critical information necessary to evaluate the appropriateness of the two features related by Assembly_joint to their use in the assembly, given the process used for the joint.

NOTE 3 Assembly_joint provides critical information necessary to evaluate the appropriateness of the relative position of the two features related by Assembly_joint, given the process used for the joint, and the technology used for each of the associated components, including related tolerances.

NOTE 4 One of these components may be an Interconnect_module_component (see 4.2.442).

The data associated with an Assembly_joint are the following:

- assembly_features;
- associated_assembly;
- description;
- name.



Note - the insulated wires are each an Assembly_component that is a Routed_interconnect_component and is also an Interface_component

Figure 8 – Assembly_joint

4.2.41.1 assembly_features

The assembly_features specify exactly two members of any of the following: Assembly_module_component_terminal (see 4.2.45), Bare_die_component_terminal (see 4.2.62), Interconnect_component_interface_terminal (see 4.2.438), Interconnect_component_join_terminal (see 4.2.439), Interconnect_module_component_terminal (see 4.2.445), Package_terminal_occurrence (see 4.2.583), Packaged_component_join_terminal (see 4.2.586), or Packaged_connector_component_interface_terminal (see 4.2.589), for the Assembly_joint.

4.2.41.2 associated_assembly

The associated_assembly specifies the Assembly_module (see 4.2.43) or the Assembly_group_component (see 4.2.40) that contains the Assembly_joint.

4.2.41.3 description

The description specifies the text that in human-interpretable form describes the Assembly_joint. The description need not be specified for a particular Assembly_joint.

4.2.41.4 name

The name specifies the string of text that identifies the Assembly_joint.

Constraints:

WR1: A feature shall not be assembled to itself either directly or indirectly.

WR2: Assembly_features_1 shall be in the associated_assembly.

WR3: Assembly_features_2 shall be in the associated_assembly.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_joint
  SUPERTYPE OF (assembled_with_fasteners ANDOR assembled_with_bonding)
  SUBTYPE OF (managed_design_object);
  assembly_features_1 : assembled_feature_select;
  assembly_features_2 : assembled_feature_select;
  associated_assembly : assembly_module_or_assembly_group_component;
  description         : OPTIONAL STRING;
  name                 : STRING;
UNIQUE
  UR1: name, associated_assembly;
WHERE
  WR1: assembly_features_1 :&#60;&#62;: assembly_features_2;
  WR2: ('AP210_ARM.ASSEMBLED_WITH_FASTENERS' IN TYPEOF(SEL)) OR
      (('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(assembly_features_1)) AND
      (SIZEOF(QUERY( ay &#60;*
        assembly_features_1\component_feature.associated_component\
        assembly_component.assembly | ay.assembly &#60;&#62;
        associated_assembly)) = 0));
  WR3: ('AP210_ARM.ASSEMBLED_WITH_FASTENERS' IN TYPEOF(SEL)) OR
      (('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(assembly_features_2)) AND
      (SIZEOF(QUERY( ay &#60;*
        assembly_features_2\component_feature.associated_component\
        assembly_component.assembly | ay.assembly &#60;&#62;
        associated_assembly)) = 0));
END_ENTITY; -- assembly_joint

( *
( *

```

4.2.42 Assembly_material_composition_relationship

An Assembly_material_composition_relationship is a type of Managed_design_object (see 4.2.524). An Assembly_material_composition_relationship is the association between a Physical_unit_design_view (see 4.2.667) and Ee_material (see 4.2.318) for the purpose of bulk material callout.

NOTE This application object is intended solely to support raw material callout whose shape is not explicitly modeled in the product model.

EXAMPLE 1 Solder identified as a bonding agent may be aggregated for the assembly using this Application Object.

The data associated with an `Assembly_material_composition_relationship` are the following:

- `as_required`;
- `assembly`;
- `component_material`;
- `quantity`;
- `security_code`.

4.2.42.1 `as_required`

The `as_required` specifies the LOGICAL type for the `Assembly_material_composition_relationship`. This type permits a user to specify that the amount of material to be supplied with the product may be determined at assembly on an item by item basis, where the item referred to is a real product delivered to a customer.

EXAMPLE 1 Adhesives, insulation and wire may all be cases where the value of the attribute is TRUE.

4.2.42.2 `assembly`

The `assembly` specifies the role of the `Physical_unit_design_view` (see 4.2.667) for the `Assembly_material_composition_relationship`.

4.2.42.3 `component_material`

The `component_material` specifies the role of the `Material_definition` (see 4.2.531) for the `Assembly_material_composition_relationship`.

4.2.42.4 `quantity`

The `quantity` specifies the role of the `Data_element` (see 4.2.211) for the `Assembly_material_composition_relationship`. The quantity need not be specified for the `Assembly_material_composition_relationship`.

4.2.42.5 `security_code`

The `security_code` specifies the role of the `Security_classification` (see 4.2.804) for the `Assembly_material_composition_relationship`. The `security_code` is optional and need not be supplied.

Constraints:

WR1: For an occurrence of a bulk material product, units shall be specified as one of the following: length_unit; amount_of_substance_unit; mass_unit; volume_unit; area_unit.

IP1: Wire used for jumpers that are explicitly defined shall be specified as a member of Interconnect_component.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_material_composition_relationship
  SUBTYPE OF (managed_design_object);
    quantity          : OPTIONAL data_element;
    assembly           : physical_unit_design_view;
    component_material : material_definition;
    security_code      : OPTIONAL security_classification;
    as_required        : LOGICAL;
END_ENTITY; -- assembly_material_composition_relationship
(*
```

4.2.43 Assembly_module

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module is a type of Physical_unit_design_view (see 4.2.667). An Assembly_module may be a Pca (see 4.2.639). An Assembly_module is an assembly design view of a product. This application object supports design reuse, modular product definitions, component libraries, and related technologies. There may be discipline views that further specialize this view.

EXAMPLE 1 Figure 80 is an illustration of an Assembly_module.

NOTE 1 An Assembly_module could be a ball grid array design, an MCM design, a disk drive design, a line card or plug-in module in a telecommunications rack, or a line replaceable unit in an airframe or spacecraft. Incomplete designs that have some specified functionality as defined by a functional definition may be included in a component library application, for example. The designs may be included in an active design by reference. At the appropriate time in the design phase, the single reference is replaced with individual references to the elements that compose the Assembly_module. Support is provided for maintaining the information necessary to trace back to the original Assembly_module from the released Assembly_component (see 4.2.34) through the Component_make_from_relationship (see 4.2.126).

NOTE 2 Product categories other than Pca, such as MCM-L (Multichip Module, Laminated), MCM-D (Multichip Module, Deposited) and MCM-C (Multichip Module, Ceramic) may be included through the use of this application object, and population of the Technology (see 4.2.886) and Ee_product_category (see 4.2.320) application objects.

EXAMPLE 2 In this example (based on the previous notes), assembly technology is the process that bonds two products together (e.g., soldering, welding) by joining individual features of one product to individual features of the other product. The Assembly_module would have two components: A Packaged_part (see 4.2.591) and an Interconnect_module. The Interconnect_module would include a Land (see 4.2.490) pattern designed to optimize the assembly process yield for the Packaged_part and the Stratum (see 4.2.845) stackup of the Interconnect_module combination. Full knowledge of the Interconnect_module technology is available through the complete definition of the Passage_technology (see 4.2.638) and Stratum_technology (see 4.2.871) supported in this part of ISO 10303. The assembly technology characteristics assigned by an enterprise may be exchanged through population of the Ee_product_category and Ee_product_related_product_category (see 4.2.326) application objects. Technology classification may be exchanged using the Technology application object. If desired, Analytical_representation (see 4.2.22) may be populated to support process simulation. Package (see 4.2.573) may be used instead of Packaged_part in the Assembly_module if there is no interest in electrical characteristics.

Constraints:

WR1: A Laminate_component shall not be in the role of component for the Assembly_composition_-relationship that references an Assembly_module in the role of assembly

NOTE EXPRESS specification:

```
*)
ENTITY assembly_module
    SUPERTYPE OF (ONEOF(pca,
                        reference_packaged_part_assembly_implementation))
    SUBTYPE OF (physical_unit_design_view);
END_ENTITY; -- assembly_module

( *
```

4.2.44 Assembly_module_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_component is a type of Assembly_component (see 4.2.34). This application object is an occurrence of the usage view of an Assembly_module (see 4.2.43), and could be either a design reuse item or an occurrence of the usage view of a real item.

NOTE Assembly_module_component represents an occurrence of a part concept for usage in one or more design views for the same product, where each design view is for a different version of the product. This is useful in a vertically integrated concurrent engineering environment, where full data sharing is allowed, as it permits visibility into the fact that there is a design definition of the component that is an Assembly_module.

EXAMPLE 1 It is useful to consider a daughter board instance as an Assembly_module_component rather than as an instance of Packaged_part (see 4.2.591) in this part of ISO 10303 for traceability to the design view for archival purposes for the design organization.

The data associated with an `Assembly_module_component` are the following:

- `access_mechanisms`;
- `usage_definition`.

4.2.44.1 `access_mechanisms`

The `access_mechanisms` is an inverse relationship that specifies that the existence of the `Assembly_module_component` is dependent on the existence of the `Assembly_module_component_terminal` (see 4.2.45) that specifies the `Assembly_module_component` as its `associated_component`. There shall be two or more `Assembly_module_component_terminals` for an `Assembly_module_component`.

4.2.44.2 `usage_definition`

The `usage_definition` specifies the role of the `Assembly_module_usage_view` (see 4.2.51) for the `Assembly_module_component`.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_component
--tt tc1 wd9 modified
    SUPERTYPE OF(assembly_module_macro_component)
--tt tc1 wd11 modified
    SUBTYPE OF (physical_component);
    SELF\assembly_component.usage_definition : assembly_module_usage_view;
    INVERSE
        access_mechanisms : SET [2:?] OF assembly_module_component_terminal FOR
                                associated_component;
END_ENTITY; -- assembly_module_component

( *
```

4.2.45 `Assembly_module_component_terminal`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Assembly_module_component_terminal` is a type of `Component_terminal` (see 4.2.137). The `Assembly_module_component_terminal` is included with the `Assembly_module_component` to assure the data integrity of references to the terminals of the `Assembly_module_component`.

The data associated with an `Assembly_module_component_terminal` are the following:

- `associated_component`;
- `definition`.

4.2.45.1 associated_component

The associated_component specifies the role of the Assembly_module_component (see 4.2.44) for the Assembly_module_component_terminal.

4.2.45.2 definition

The definition specifies the role of the Assembly_module_terminal (see 4.2.50) for the Assembly_module_component_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_module_component_terminal
  SUBTYPE OF (physical_component_terminal);
    SELF\component_feature.definition          : assembly_module_terminal;
    SELF\component_feature.associated_component : assembly_module_component;
END_ENTITY; -- assembly_module_component_terminal

--tt tc1 wd9 added
(*
```

4.2.46 Assembly_module_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_interface_terminal is a type of Assembly_module_terminal (see 4.2.50).

NOTE EXPRESS specification:

```
*)
ENTITY assembly_module_interface_terminal
  SUBTYPE OF (assembly_module_terminal);
END_ENTITY; -- assembly_module_interface_terminal

--tt tc1 wd9 added
(*
```

4.2.47 Assembly_module_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_join_terminal is a type of Assembly_module_terminal (see 4.2.50).

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_join_terminal
  SUBTYPE OF (assembly_module_terminal);
INVERSE
--tt tc1 wd9 modified
defining_terminal : component_feature_to_physical_usage_view_assignment for
  defined_part_feature;
WHERE
  WR1: 'AP210_ARM.INTERCONNECT_MODULE_JOIN_TERMINAL' IN TYPEOF(defining_terminal.assigned_
  WR2: 'AP210_ARM.COMPONENT_TERMINAL_TO_ASSEMBLY_MODULE_JOIN_TERMINAL_ASSIGNMENT' IN TYPEOF
END_ENTITY; -- assembly_module_join_terminal

--tt tc1 wd8 added
(*

```

4.2.48 Assembly_module_macro_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_macro_component is a type of Assembly_component (see 4.2.34).

The data associated with an Assembly_module_macro_component are the following:

- access_mechanisms;
- usage_definition;
- usage_view_definition.

4.2.48.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Assembly_module_macro_component is dependent on the existence of the Assembly_module_component_terminal (see 4.2.45) that specifies the Assembly_module_macro_component as its associated_component. There shall be two or more Assembly_module_component_terminal for an Assembly_module_macro_component.

4.2.48.2 usage_definition

The usage_definition specifies the role of an Assembly_module (see 4.2.43) for the Assembly_module_macro_component.

4.2.48.3 usage_view_definition

The usage_view_definition is a derived role of an Assembly_module_usage_view (see 4.2.51) which specifies the Assembly_module_usage_view that is identified via the path SELF\assembly_component.usage_definition\physical_unit_design_view.usage_view.

Constraints:

WR1: The Assembly_module_usage_view (see 4.2.51) referenced by the usage_view_definition shall be the same Assembly_module_usage_view referenced by the associated_definition which is referenced by the definition which is referenced by the access_mechanisms.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_macro_component
--tt tc1 wd9 modified
    SUBTYPE OF (assembly_module_component);
--tt tc1 wd9 modified
    design_definition : assembly_module;
--tt tc1 wd9 modified
    INVERSE
        join_access_mechanisms : SET [1:?] OF assembly_module_macro_component_join_terminal FOR
            associated_component;
WHERE
    WR1: SIZEOF (QUERY( ammcjt &#60;* join_access_mechanisms |
        ammcjt\component_feature.definition\part_feature.associated_definition :&#60;
            design_definition\physical_unit_design_view.usage_view)) = 0;
END_ENTITY; -- assembly_module_macro_component

--tt tc1 wd9 added
( *
```

4.2.49 Assembly_module_macro_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_macro_component_join_terminal is a type of Assembly_module_component_terminal (see 4.2.45).

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_macro_component_join_terminal
  SUBTYPE OF (assembly_module_component_terminal);
    SELF\component_feature.definition      : assembly_module_join_terminal;
    SELF\component_feature.associated_component : assembly_module_macro_component;
END_ENTITY; -- assembly_module_macro_component_join_terminal

( *

```

4.2.50 Assembly_module_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_terminal is a type of Part_terminal (see 4.2.630). An Assembly_module_terminal may be a Pca_terminal (see 4.2.640). An Assembly_module_terminal may be associated with an instance of Assembly_module_usage_view_connector_relationship (see 4.2.52) to specify connector level interface definition for an assembly. This association is maintained in the usage view. If available, the terminal_connection_zone shall be supplied.

NOTE The Assembly_module_terminal supports multi-level design, analysis, and verification in a multi-discipline environment. Explicit description of assembly_module_terminals allows connectivity requirements to be explicitly stated and interference requirements to be identified early in the design and development cycle.

The data associated with an Assembly_module_terminal are the following:

- associated_definition;
- feature_shape;
- reference_terminal;
- related_connector;
- terminal_connection_zone.

4.2.50.1 associated_definition

The associated_definition specifies the role of the Assembly_module_usage_view (see 4.2.51) for the Assembly_module_terminal.

4.2.50.2 feature_shape

The feature_shape specifies the role of the Feature_shape_occurrence (see ??) for the Assembly_module_terminal. There shall be one or more Feature_shape_occurrence for an Assembly_module_terminal.

4.2.50.3 reference_terminal

The reference_terminal specifies a Packaged_part_interface_terminal (see 4.2.592) for the Assembly_module_terminal. A reference_terminal need not be specified for a Package_part_interface_terminal.

4.2.50.4 related_connector

The related_connector specifies an Assembly_module_usage_view_connector_relationship (see 4.2.52) for the Assembly_module_terminal. A related_connector need not be specified for a Package_part_interface_terminal.

4.2.50.5 terminal_connection_zone

The terminal_connection_zone specifies the role of the Connection_zone (see 4.2.185) for the Assembly_module_terminal. There may be more than one Connection_zone for an Assembly_module_terminal. The terminal_connection_zone is optional and need not be supplied. If the data is available in the source system, it shall be supplied.

NOTE The Feature_shape_occurrence (see ??) specified by the terminal or mounting feature may be the same Feature_shape_occurrence that is specified by the associated Connection_zone (see 4.2.185). In this case the whole terminal is the Connection_zone.

Constraints:

IP1: If a related_connector is specified, the position, shape, and orientation of the Assembly_module_terminal shall be overridden by the reference_terminal position, shape, and orientation.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_terminal
  SUPERTYPE OF ((ONEOF(
    assembly_module_interface_terminal,
    assembly_module_join_terminal))
    ANDOR (pca_terminal))
  SUBTYPE OF (part_terminal);
  SELF\part_feature.associated_definition : assembly_module_usage_view;
  reference_terminal :
OPTIONAL packaged_part_interface_terminal;
  related_connector :
OPTIONAL assembly_module_usage_view_connector_relationship;
  terminal_connection_zone : OPTIONAL SET [1:?] OF connection_zone;
INVERSE
  SELF\part_feature.feature_shape : SET [1:?] OF usage_concept_usage_relationship;
  associated_usage;
END_ENTITY; -- assembly_module_terminal
( *

```

4.2.51 Assembly_module_usage_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_usage_view is a type of Physical_unit_usage_view (see 4.2.672). An Assembly_module_usage_view may be a Pca_usage_view (see 4.2.641). An Assembly_module_usage_view provides additional domain knowledge for the design organization.

NOTE 1 The Assembly_module_usage_view supports intellectual property protection. A design organization can control the amount of data they provide to a licensee or purchaser by including only a specifically tailored set of data in a usage view.

NOTE 2 Shape information is provided by Physical_unit_3d_shape (see 4.2.666) or by Physical_unit_planar_shape (see 4.2.670).

EXAMPLE 1 This data can include the information currently included in an interface control drawing (ICD).

EXAMPLE 2 Figure 9 illustrates that the information in an Assembly_module_usage_view can be different from the information in a design view. The information in this illustration is sufficient to analyse airflow characteristics and interference characteristics, as well as information related to the connectors in the design.

The data associated with an `Assembly_module_usage_view` are the following:

- `access_mechanisms`;
- `implemented_function`;
- `maximum_negative_component_height`;
- `maximum_positive_component_height`.

4.2.51.1 access_mechanisms

The `access_mechanisms` is an inverse relationship that specifies that the existence of the `Assembly_module_usage_view` is dependent on the existence of the `Assembly_module_terminal` (see 4.2.50) that specifies the `Assembly_module_usage_view` as its `associated_definition`. There shall be two or more `Assembly_module_terminals` for an `Assembly_module_usage_view`.

4.2.51.2 implemented_function

The `implemented_function` specifies the role of the `Functional_unit_usage_view` (see 4.2.410) for the `Assembly_module_usage_view`.

4.2.51.3 maximum_negative_component_height

The `maximum_negative_component_height` specifies the largest `Datum_based_length_measure` (see 4.2.219) in the negative direction from the `Datum` (see 4.2.214) for the `Assembly_module_usage_view`. The `maximum_negative_component_height` need not be specified for a particular `Assembly_module_usage_view`.

EXAMPLE 1 An example of the `maximum_negative_component_height` attribute is for a printed circuit assembly with through-hole components, where the maximum lead protrusion dimension is given. That dimension is the `maximum_negative_component_height`. Figure 9 illustrates this example. Another example of the `maximum_negative_component_height` attribute is for a printed circuit assembly with surface mounted components on both sides. The `maximum_negative_component_height` attribute is for the distance between the established datum and the maximum material condition of the vertical dimension of the highest surface mount component on the secondary side of the printed circuit assembly.

4.2.51.4 maximum_positive_component_height

The `maximum_positive_component_height` specifies the largest `Datum_based_length_measure` (see 4.2.219) in the positive direction from the `Datum` (see 4.2.214) for the `Assembly_module_usage_view`. The `maximum_positive_component_height` need not be specified for a particular `Assembly_module_usage_view`.

EXAMPLE 1 An example of the `maximum_positive_component_height` attribute is for a printed circuit assembly where the maximum component height in the positive direction relative to a datum is given. This dimension pertains to the fit of the printed circuit assembly in its next assembly. That dimension is the `maximum_positive_component_height`. Figure 9 illustrates this example.

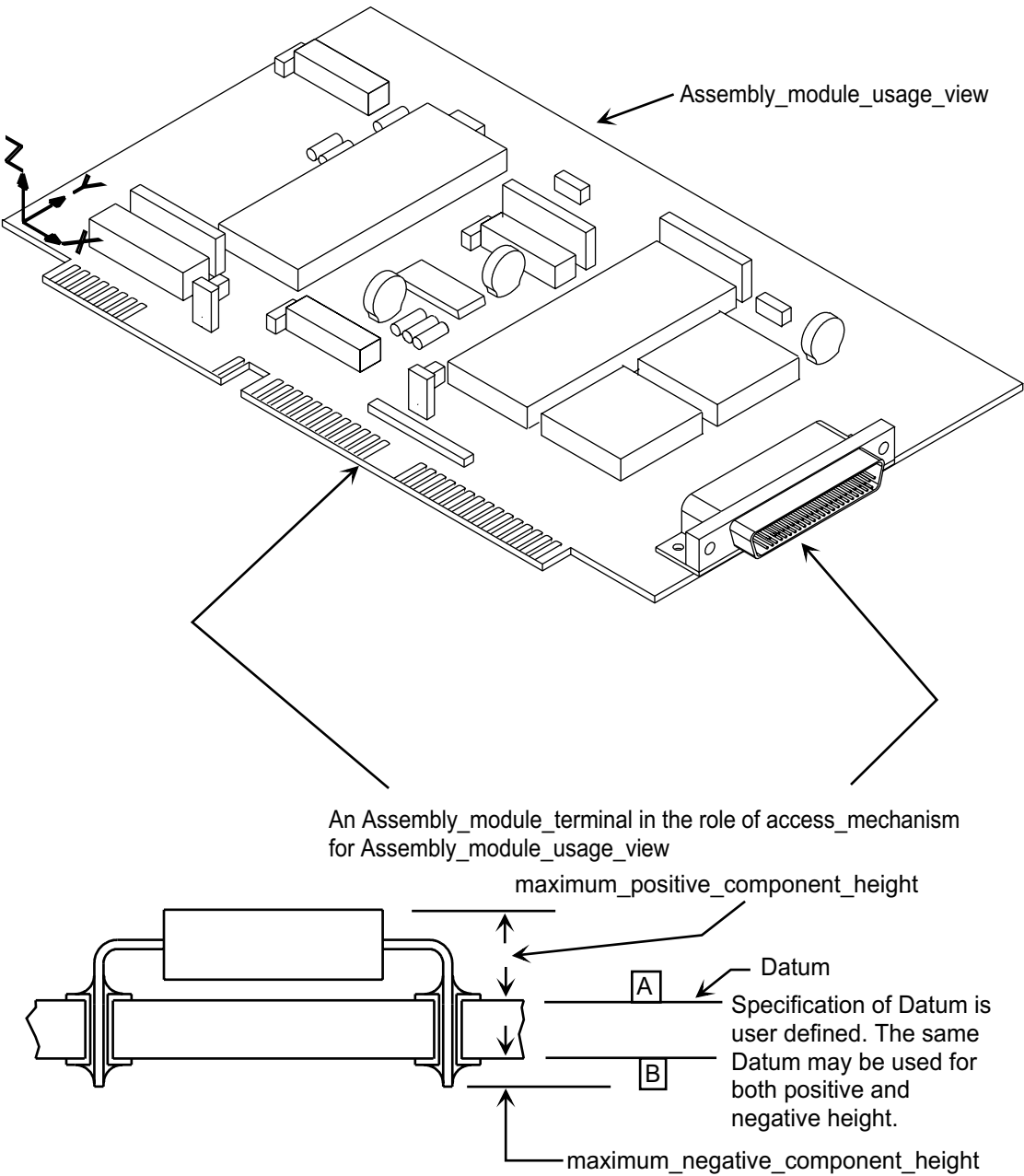


Figure 9 – Assembly_module_usage_view

Constraints:

WR1: The direction from the Datum for the maximum_negative_component_height shall be negative in the Z axis direction.

WR2: The direction from the Datum for the maximum_positive_component_height shall be positive in the Z axis direction.

WR3: The function implemented shall be the function whose terminals are mapped to the terminals of this assembly module usage view.

IP1: The Datum orientation vector direction shall be considered to be positive when it is parallel to the normal of the top side of the interconnect substrate instance.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_usage_view
  SUPERTYPE OF (pca_usage_view)
  SUBTYPE OF (physical_unit_usage_view);
  implemented_function          : functional_unit_usage_view;
  maximum_negative_component_height : OPTIONAL datum_based_length_measure;
  maximum_positive_component_height : OPTIONAL datum_based_length_measure;
INVERSE
  access_mechanisms : SET [2:?] OF assembly_module_terminal FOR
                        associated_definition;
END_ENTITY; -- assembly_module_usage_view

( *
```

4.2.52 Assembly_module_usage_view_connector_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_module_usage_view_connector_relationship is a type of Usage_concept_usage_relationship (see 4.2.942) that associates the placement and orientation of a Packaged_connector (see 4.2.587) with an Assembly_module_usage_view (see 4.2.51). An Assembly_module_usage_view_connector_relationship supports the identification of connector placement data, when the design details are not available.

The data associated with an Assembly_module_usage_view_connector_relationship are the following:

— externally_visible_partial_reference_designation.

4.2.52.1 externally_visible_partial_reference_designation

An externally_visible_partial_reference_designation specifies an Ee_name for an Assembly_module_usage_view_connector_relationship (see 4.2.52). An externally_visible_partial_reference_designation supports the identification of a connector as a partial reference designation independent of the internal reference designation assigned by the designer of the product. This independence is required when the connector is actually designed as part of a complete lower level assembly.

Constraints:

UR1: The combination of externally_visible_partial_reference_designation and associating_usage inherited from Usage_concept_usage_relationship shall be unique in a population of Assembly_module_usage_view_connector_relationship.

IP1: The associating_usage inherited from Usage_concept_usage_relationship shall be an Assembly_module_usage_view.

IP2: The associated_usage inherited from Usage_concept_usage_relationship shall be a Packaged_connector.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_module_usage_view_connector_relationship
  SUBTYPE OF (usage_concept_usage_relationship);
    associating_usage : assembly_module_usage_view;
    SELF\usage_concept_usage_relationship.associated_usage : packaged_connector;
    externally_visible_partial_reference_designation : ee_name;
UNIQUE
  UR1 : externally_visible_partial_reference_designation,
  associating_usage;
END_ENTITY; -- assembly_module_usage_view_connector_relationship

( *
```

4.2.53 Assembly_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_requirement_allocation is a type of Managed_design_object (see 4.2.524). An Assembly_requirement_allocation may be one or more of an Assembly_separation_requirement_allocation (see 4.2.54), an Electromagnetic_compatibility_requirement_allocation (see 4.2.347), or a Thermal_requirement_allocation (see 4.2.907). An Assembly_requirement_allocation is the relationship between the Ee_requirement_occurrence (see 4.2.330) and either an Assembly_component (see 4.2.34) or a Component_feature (see 4.2.118) that satisfies that Ee_requirement_occurrence.

The data associated with an Assembly_requirement_allocation are the following:

- assembly_assignment;
- requirement.

4.2.53.1 assembly_assignment

The assembly_assignment specifies either an Assembly_component (see 4.2.34) or a Component_feature (see 4.2.118) for the Assembly_requirement_allocation.

4.2.53.2 requirement

The requirement specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Assembly_requirement_allocation.

NOTE EXPRESS specification:

```

*)
ENTITY assembly_requirement_allocation
  SUPERTYPE OF ((thermal_requirement_allocation ANDOR
    electromagnetic_compatibility_requirement_allocation) ANDOR
    assembly_separation_requirement_allocation)
  SUBTYPE OF (managed_design_object);
  assembly_assignment : component_or_feature;
  requirement         : ee_requirement_occurrence;
END_ENTITY; -- assembly_requirement_allocation

( *
```

4.2.54 Assembly_separation_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Assembly_separation_requirement_allocation is a type of Assembly_requirement_allocation (see 4.2.53). An Assembly_separation_requirement_allocation provides for the physical spacing required among components of the assembly. The Assembly_separation_requirement_allocation is for the purpose of specifying separation between items based on the Ee_requirement_occurrence (see 4.2.330).

NOTE Multiple members of this entity may be populated in order to specify all the sets with which a given set will be disjoint, either thermally or electromagnetically.

EXAMPLE 1 A typical usage of this AO is to reference a member of Layout_spacing_requirement_occurrence (see 4.2.509) as the requirement inherited from Assembly_requirement_allocation.

The data associated with an `Assembly_separation_requirement_allocation` are the following:

— `disjoint_assignment`.

4.2.54.1 `disjoint_assignment`

The `disjoint_assignment` specifies either an `Assembly_component` (see 4.2.34) or a `Component_feature` (see 4.2.118) for the `Assembly_separation_requirement_allocation`. There shall be one or more `Assembly_component` or `Component_feature` for the `Assembly_separation_requirement_allocation`. The `disjoint_assignment` shall be applied pairwise.

Constraints:

IP1: The value of `Length_data_element` (see 4.2.511) shall be specified as a characteristic of the `Ee_requirement_occurrence` (see 4.2.330) from `Assembly_requirement_allocation`.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_separation_requirement_allocation
  SUBTYPE OF (assembly_requirement_allocation);
    disjoint_assignment : SET [1:?] OF component_or_feature;
END_ENTITY; -- assembly_separation_requirement_allocation
( *
```

4.2.55 `Assembly_technology_specification`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Assembly_technology_specification` is a type of `Process_specification` (see 4.2.740) that identifies and describes an activity to be performed on an `Ee_product` (see 4.2.319) or `Ee_material` (see 4.2.318) in order to produce an assembly.

NOTE EXPRESS specification:

```
*)
ENTITY assembly_technology_specification
  SUBTYPE OF (process_specification);
END_ENTITY; -- assembly_technology_specification
( *
```

4.2.56 `Axis_placement`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Axis_placement` is a type of `Shape_item` (see 4.2.818). An `Axis_placement` may be an `Axis_placement_2d` (see 4.2.57) or an `Axis_placement_3d` (see 4.2.58). An `Axis_placement` is the transformation and orientation of one `Cartesian_coordinate_system` (see 4.2.86) relative to a referenced coordinate system. An `Axis_placement` is the rigid body transformation that transforms the geometry of the object being placed from its coordinate system to the `Cartesian_coordinate_system` in which the object is being placed. The dimensionality of `Axis_placement` is determined by the dimensionality of the object being placed. The interpretation of this Application Object shall be to an `axis2_placement_3d` or to an `axis2_placement_2d`.

The data associated with an `Axis_placement` are the following:

- `reference_direction`;
- `translation`.

4.2.56.1 `reference_direction`

The `reference_direction` specifies a `Direction` (see 4.2.304) as a reference for an occurrence of a physical device in an `Axis_placement`.

4.2.56.2 `translation`

The `translation` specifies the `Cartesian_point` (see 4.2.87) that is the origin of a `Cartesian_coordinate_system` (see 4.2.86) relative to the origin of a reference `Cartesian_coordinate_system` while maintaining the angular relationship among the axes of the two coordinate systems in the `Axis_placement`.

NOTE EXPRESS specification:

```
*)
ENTITY axis_placement
  SUPERTYPE OF (ONEOF (axis_placement_3d,axis_placement_2d))
  SUBTYPE OF (shape_item);
    reference_direction : direction;
    translation         : cartesian_point;
END_ENTITY; -- axis_placement
(*
```

4.2.57 `Axis_placement_2d`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Axis_placement_2d` is a type of `Axis_placement` (see 4.2.56) that has two dimensions of translation.

NOTE EXPRESS specification:

```
*)
ENTITY axis_placement_2d
  SUBTYPE OF (axis_placement);
END_ENTITY; -- axis_placement_2d
```

(*

4.2.58 Axis_placement_3d

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Axis_placement_3d is a type of Axis_placement (see 4.2.56) that has three dimensions of translation.

NOTE EXPRESS specification:

```
*)
ENTITY axis_placement_3d
  SUBTYPE OF (axis_placement);
END_ENTITY; -- axis_placement_3d
```

--tt tc1 wd9 modified

(*

4.2.59 Bare_die

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die is a type of a Physical_unit_usage_view (see 4.2.672). A Bare_die is used to describe the physical shape and function of a semiconductor device.

NOTE 1 In the PCA domain an occurrence of a Bare_die may be installed into an occurrence of a Package (see 4.2.573) to realize a Packaged_part (see 4.2.591), an occurrence of which is then mounted in the assembly. In this role of performing a function, a Bare_die has terminals.

NOTE 2 In the PCB domain an occurrence of a Bare_die may be installed into the design of an Interconnect_module (see 4.2.440) to realize an embedded component. In this role, the external surfaces of the Bare_die will be important as they form part of the location information for the occurrence.

NOTE 3 A Bare_die implements electrical functionality. It is not necessary to exchange Bare_die information for parts such as molded or ceramic resistors where there is not a separately identifiable and common industry understanding of a die.

NOTE 4 In the PCA domain an occurrence of a Bare_die may be mounted directly onto the occurrence of a usage view of a PCB.

NOTE 5 The existence of oxide or other thin contaminate protection is considered normal for a Bare_die.

The data associated with a Bare_die are the following:

- access_mechanisms;
- implemented_function;
- least_material_condition_centroid_location;
- maximum_height_above_die_seating_plane;
- maximum_material_condition_centroid_location;
- die_seating_plane;
- unit_technology.

EXAMPLE 1 A semiconductor die is a familiar example used in packaged components on PCAs.

4.2.59.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Bare_die is dependent on the existence of the Minimally_defined_bare_die_terminal (see 4.2.545) that specifies the Bare_die as its associated_definition. There shall be two or more Minimally_defined_bare_die_terminal for a Bare_die.

4.2.59.2 implemented_function

The implemented_function specifies the role of the Functional_unit_usage_view (see 4.2.410) for the Bare_die.

4.2.59.3 least_material_condition_centroid_location

The least_material_condition_centroid_location specifies one role of the Cartesian_point (see 4.2.87) for the Bare_die. The least_material_condition_centroid_location is the center of mass for a bare die in the minimum material state. The least_material_condition_centroid_location need not be specified for a particular Bare_die.

4.2.59.4 maximum_height_above_die_seating_plane

The maximum_height_above_die_seating_plane specifies the role of the Length_data_element (see 4.2.511) for the Bare_die. The maximum_height_above_die_seating_plane is the largest extent of the Bare_die above the die_seating_plane. The maximum_height_above_die_seating_plane need not be specified for a particular Bare_die.

4.2.59.5 maximum_material_condition_centroid_location

The maximum_material_condition_centroid_location specifies one role of the Cartesian_point (see 4.2.87) for the Bare_die. The maximum_material_condition_centroid_location is the center of mass for a bare die in the state that contains the most material. The maximum_material_condition_centroid_location need not be specified for a particular Bare_die.

4.2.59.6 die_seating_plane

The die_seating_plane is an inverse relationship that specifies that the existence of the Bare_die is dependent on the existence of the Seating_plane (see 4.2.802) that is also a standard orientation reference for assembly purposes for the Bare_die. The seating_plane is simulating a mounting surface.

4.2.59.7 unit_technology

The unit_technology specifies the role of the Technology (see 4.2.886) for the Bare_die. This part of ISO 10303 does not predefined unit_technology classification.

EXAMPLE 1 Examples include a classification for process and material such as CMOS, BICMOS, ECL, etc.

NOTE 1 Additional categorization is supported with this part of ISO 10303 by using Ee_product_category (see 4.2.320) and related application objects.

NOTE 2 The Bare_die is a monolithic (silicon, germanium, gallium arsenide, etc.) component before encapsulation in a package. It has generally been passivated (e.g., with an oxide layer) to protect it from contamination. Packaging is not required in order for a Bare_die description to be exchanged in conformance to this standard.

Constraints:

WR1: The function implemented shall be the function whose terminals are mapped to the terminals of the bare die.

NOTE This is a test of express in a note

EXPRESS specification:

```

*)
ENTITY bare_die
  SUBTYPE OF (physical_unit_usage_view);
    least_material_condition_centroid_location : OPTIONAL cartesian_point;
    maximum_height_above_seating_plane         : OPTIONAL length_data_element;
    maximum_material_condition_centroid_location : OPTIONAL cartesian_point;
    SELF\physical_unit.unit_technology          : technology;
    implemented_function                        : functional_unit_usage_view;
  INVERSE
    access_mechanisms : SET [2:?] OF minimally_defined_bare_die_terminal FOR
                          associated_definition;
    die_seating_plane : non_feature_shape_element FOR
                          scope;
  UNIQUE
    UR1 : implemented_function, unit_technology;
WHERE
  WR1: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'BARE_DIE_TOP_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR2: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'BARE_DIE_BOTTOM_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR3: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'BARE_DIE_EDGE_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR4: 'AP210_ARM.SEATING_PLANE' IN TYPEOF(die_seating_plane);
END_ENTITY; -- bare_die
( *

```

4.2.60 Bare_die_bottom_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_bottom_surface is a type of Part_mounting_feature (see 4.2.618).

The data associated with a Bare_die_bottom_surface are the following:

— associated_definition.

4.2.60.1 associated_definition

The associated_definition specifies the role of the Bare_die (see 4.2.59) for the Bare_die_bottom_surface.

NOTE EXPRESS specification:

```
*)
ENTITY bare_die_bottom_surface
  SUBTYPE OF (bare_die_surface);
  SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_bottom_surface

( *
```

4.2.61 Bare_die_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_component is a type of Assembly_component (see 4.2.34) that is defined by a Bare_die (see 4.2.59).

The data associated with a Bare_die_component are the following:

— usage_definition.

4.2.61.1 usage_definition

The usage_definition specifies the role of the Bare_die for the Bare_die_component.

NOTE EXPRESS specification:

```
*)
ENTITY bare_die_component
--tt tc1 wd11 modified
  SUBTYPE OF (physical_component);
  SELF\assembly_component.usage_definition : bare_die;
END_ENTITY; -- bare_die_component

( *
```

4.2.62 Bare_die_component_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_component_terminal is a type of Component_terminal (see 4.2.137) that is an instance of a Bare_die_terminal (see 4.2.66). The Bare_die_component_terminal is the area on the bare die instance onto which an enterprise will place bond wires, joining the active areas on the die instance to the next level of assembly. It is not necessarily the shape of the land itself on the die.

The data associated with a Bare_die_component_terminal are the following:

- associated_component;
- definition.

4.2.62.1 associated_component

The associated_component specifies the Bare_die_component (see 4.2.61) to which the Bare_die_component_terminal provides electrical access.

4.2.62.2 definition

The definition specifies the role of the Bare_die_terminal for the Bare_die_component_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY bare_die_component_terminal
  SUBTYPE OF (physical_component_terminal);
    SELF\component_feature.definition          : bare_die_terminal;
    SELF\component_feature.associated_component : bare_die_component;
END_ENTITY; -- bare_die_component_terminal

--tt tc1wd9 added
(*
```

4.2.63 Bare_die_edge_segment_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_edge_segment_surface is a type of Bare_die_surface (see 4.2.65).

NOTE EXPRESS specification:

```

*)
ENTITY bare_die_edge_segment_surface
  SUBTYPE OF (bare_die_surface);
    composed_surface : bare_die_edge_surface;
    start_vertex      : edge_segment_vertex;
    end_vertex        : edge_segment_vertex;
    SELF\part_feature.associated_definition : bare_die;
WHERE
  WR1: start_vertex :=&#60;&#62;: end_vertex;
  WR2: SELF\part_feature.associated_definition :=:
    composed_surface\part_feature.associated_definition;
END_ENTITY; -- bare_die_edge_segment_surface

--tt tc1wd9 added
(

```

4.2.64 Bare_die_edge_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_edge_surface is a type of Bare_die_surface (see 4.2.65).

NOTE EXPRESS specification:

```

*)
ENTITY bare_die_edge_surface
  SUBTYPE OF (bare_die_surface);
    SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_edge_surface

--tt tc1wd9 added
(

```

4.2.65 Bare_die_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_surface is a type of Part_mounting_feature (see 4.2.618). A Bare_die_surface is one of a Bare_die_top_surface (see 4.2.68), a Bare_die_bottom_surface (see 4.2.60), a Bare_die_edge_segment_surface (see 4.2.63), or a Bare_die_edge_surface (see 4.2.64).

NOTE EXPRESS specification:

```

*)
ENTITY bare_die_surface
  ABSTRACT SUPERTYPE OF (ONEOF(
    bare_die_top_surface,
    bare_die_bottom_surface,
    bare_die_edge_segment_surface,
    bare_die_edge_surface))
  SUBTYPE OF (part_mounting_feature);
END_ENTITY; -- bare_die_surface

( *
```

4.2.66 Bare_die_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_terminal is a type of Minimally_defined_bare_die_terminal (see 4.2.545). A Bare_die_terminal is the surface area on a Bare_die (see 4.2.59) that corresponds to the functional terminal on that Bare_die.

EXAMPLE 1 A device_terminal_map in combination with a part_device_terminal are populated to map the functionality of a Bare_die to the functionality of the packaged_part for the occurrence of the Bare_die in the Packaged_part.

The data associated with a Bare_die_terminal are the following:

- external_connection_area;
- material.

4.2.66.1 external_connection_area

The external_connection_area specifies the Connection_zone (see 4.2.185) of the Bare_die_terminal. The external_connection_area is the area on the surface of a Bare_die (see 4.2.59) to which an electrically conducting material shall be attached to access the electrical functionality of the Bare_die. There may be more than one Connection_zone for a Bare_die_terminal. The external_connection_area is optional and need not be supplied if the data is unavailable in the source system. If the data is available in the source system, it shall be supplied.

NOTE The Feature_shape_occurrence (see ??) inherited by the terminal may be the same Feature_shape_occurrence that is specified by the associated Connection_zone (see 4.2.185). In this case the whole terminal is the Connection_zone.

4.2.66.2 material

The material specifies the role of the Ee_material (see 4.2.318) for the Bare_die_terminal.

Constraints:

WR1: The points comprising the external_connection_area shall reside in the shape of the Bare_die being accessed.

WR2: The points comprising the external_connection_area shall be defined in the coordinate system of the device being accessed via the external_connection_area.

NOTE EXPRESS specification:

```

*)
ENTITY bare_die_terminal
  SUBTYPE OF (minimally_defined_bare_die_terminal);
  material          : ee_material;
  external_connection_area : OPTIONAL SET [1:?] OF connection_zone;
  seating_plane_zone : OPTIONAL connection_zone_interface_plane_relationship;
END_ENTITY; -- bare_die_terminal
( *
```

4.2.67 Bare_die_terminal_surface_constituent_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_terminal_surface_constituent_relationship is a type of Shape_element_constituent_relationship (see 4.2.813).

NOTE EXPRESS specification:

```

*)
ENTITY bare_die_terminal_surface_constituent_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relate_shape_element : bare_die_surface;
  SELF\shape_element_relationship.related_shape_element : bare_die_terminal;
END_ENTITY; -- bare_die_terminal_surface_constituent_relationship

--tt tc1 wd8 add
( *
```

4.2.68 Bare_die_top_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bare_die_top_surface is a type of Part_mounting_feature (see 4.2.618).

The data associated with a Bare_die_top_surface are the following:

— associated_definition.

4.2.68.1 associated_definition

The associated_definition specifies the role of the Bare_die (see 4.2.59) for the Bare_die_top_surface.

NOTE EXPRESS specification:

```
*)
ENTITY bare_die_top_surface
  SUBTYPE OF (bare_die_surface);
  SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_top_surface

(*
```

4.2.69 Bevel_edge_feature_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bevel_edge_feature_shape is a type of Edge_shape (see 4.2.313) that describes a bevelled edge of a feature with a two-dimensional shape.

EXAMPLE 1 Figure 10 illustrates a detail of a fabrication drawing that presents some of the information provided by Bevel_edge_feature_shape. The figure illustrates the measurement conditions (including orientation) for generic width and angle attributes.

The data associated with a Bevel_edge_feature_shape are the following:

- maximum_bevel_angle;
- maximum_bevel_width;
- minimum_bevel_angle;
- minimum_bevel_width.

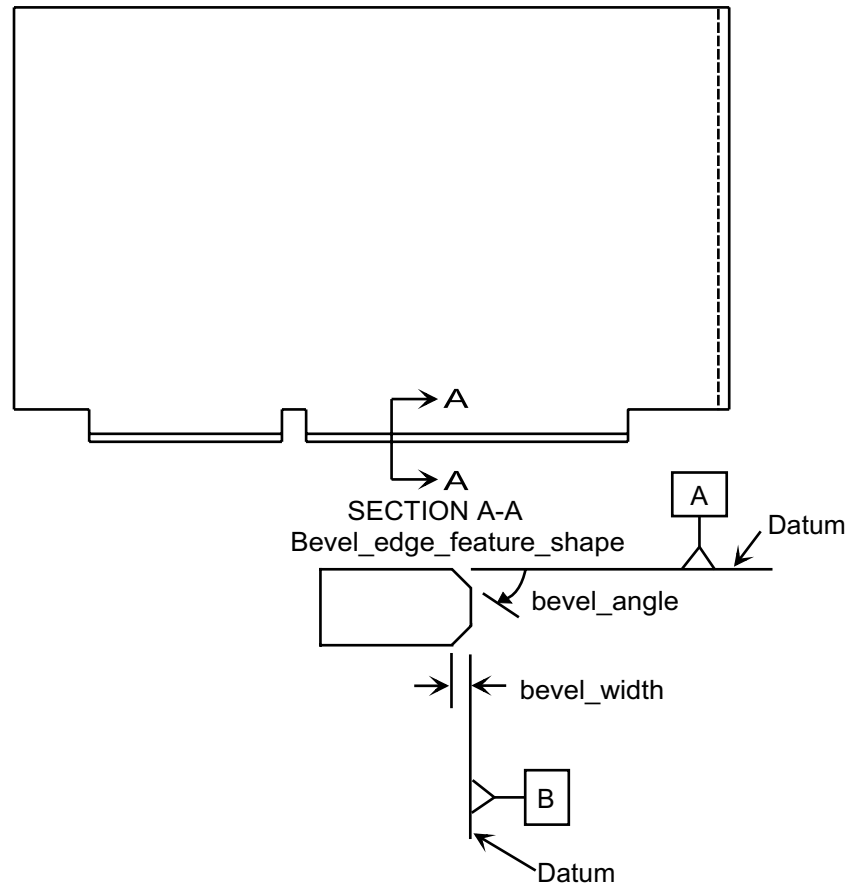


Figure 10 – Bevel_edge_feature_shape

4.2.69.1 maximum_bevel_angle

The `maximum_bevel_angle` specifies the largest positive value of an `Angle_data_element` (see 4.2.23) measured in the plane of the `Volume_shape_projection` (see ??) relative to the edge surface being beveled (Datum B in figure 10). If the `maximum_bevel_angle` is zero, no material shall be removed. If the `maximum_bevel_angle` is between zero and 90 degrees material shall be removed from the surface closest to the Stratum (see 4.2.845) whose attribute `primary_design_layer_stratum` is TRUE. If the `maximum_bevel_angle` is 90 degrees, no material shall be removed. If the `maximum_bevel_angle` is between 90 and 180 degrees, material shall be removed from the surface furthest from the Stratum whose attribute `primary_design_layer_stratum` is TRUE. `Maximum_bevel_angles` greater than 180 shall not result in material removal.

4.2.69.2 maximum_bevel_width

The `maximum_bevel_width` specifies the largest positive value of the `Length_data_element` (see 4.2.511) that is the width of the bevel relative to the edge surface that is beveled for the `Bevel_edge_feature_shape`.

4.2.69.3 minimum_bevel_angle

The `minimum_bevel_angle` specifies the smallest positive value of an `Angle_data_element` (see 4.2.23) measured in the plane of the `Volume_shape_projection` (see ??) relative to the edge surface being beveled (Datum B in figure 10). If the `minimum_bevel_angle` is zero, no material shall be removed. If the `minimum_bevel_angle` is between zero and 90 degrees material shall be removed from the surface closest to the `Stratum` (see 4.2.845) whose attribute `primary_design_layer_stratum` is TRUE. If the bevel angle is 90 degrees, no material shall be removed. If `minimum_bevel_angle` is between 90 and 180 degrees, material shall be removed from the surface furthest from `Stratum` whose attribute `primary_design_layer_stratum` is TRUE. Bevel_angles greater than 180 shall not result in material removal.

4.2.69.4 minimum_bevel_width

The `minimum_bevel_width` specifies the smallest positive value for the `Length_data_element` (see 4.2.511) that is the width of the bevel relative to the edge surface that is beveled for the `Bevel_edge_feature_shape`.

Constraints:

WR1: The `Planar_projected_shape` and `Volume_shape_projection` application objects shall be populated when this application object is populated, and shall be provided in the geometric model associated with this application object. Two members of `Datum` shall be populated when this application object is populated; one member shall be `Datum A` and the other member shall be `Datum B`.

NOTE EXPRESS specification:

```
*)
ENTITY bevel_edge_feature_shape
  SUBTYPE OF (edge_shape);
    minimum_bevel_width : length_data_element;
    minimum_bevel_angle : angle_data_element;
    maximum_bevel_width : length_data_element;
    maximum_bevel_angle : angle_data_element;
END_ENTITY; -- bevel_edge_feature_shape
```

(*

4.2.70 Blind_passage_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Blind_passage_template` is a type of `Unsupported_passage_template` (see 4.2.941). The `Blind_passage_template` includes sufficient data to determine feature properties for passages that do not penetrate the complete substrate.

The data associated with a `Blind_passage_template` are the following:

- `bottom_distance_from_datum_reference`;
- `datum_reference`;
- `of_measure_orientation`.

4.2.70.1 `bottom_distance_from_datum_reference`

The `bottom_distance_from_datum_reference` specifies the role of the `Shape_dimension_representation` (see 4.2.809) for the `Blind_passage_template`.

4.2.70.2 `datum_reference`

The `datum_reference` specifies the role of the `Stratum_surface` (see 4.2.868) for the `Blind_passage_template`.

4.2.70.3 `of_measure_orientation`

The `of_measure_orientation` specifies normal or reversed as the measure orientation for the `Blind_passage_template`. Normal specifies that the orientation is parallel to the normal of the reference surface. Reversed specifies that the orientation is anti-parallel to the normal of the reference surface.

Constraints:

IP1: A `datum_reference` shall not be a `Stratum_average_surface` (see 4.2.847).

NOTE EXPRESS specification:

```
*)
ENTITY blind_passage_template
  SUBTYPE OF (unsupported_passage_template);
    of_measure_orientation      : measure_orientation;
    datum_reference             : stratum_surface;
    bottom_distance_from_datum_reference : shape_dimension_representation;
END_ENTITY; -- blind_passage_template

( *
```

4.2.71 `Blind_via`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Blind_via is a type of Via (see 4.2.947). Each Blind_via is either a Bonded_conductive_base_blind_via (see 4.2.74), a Non_conductive_base_blind_via (see 4.2.557), or a Plated_conductive_base_blind_via (see 4.2.684) . Blind_via is defined in IEC 60050-541.

EXAMPLE 1 Figure 11 illustrates an idealized cross-section of an interconnect substrate, showing the depth of penetration of a Blind_via into the substrate.

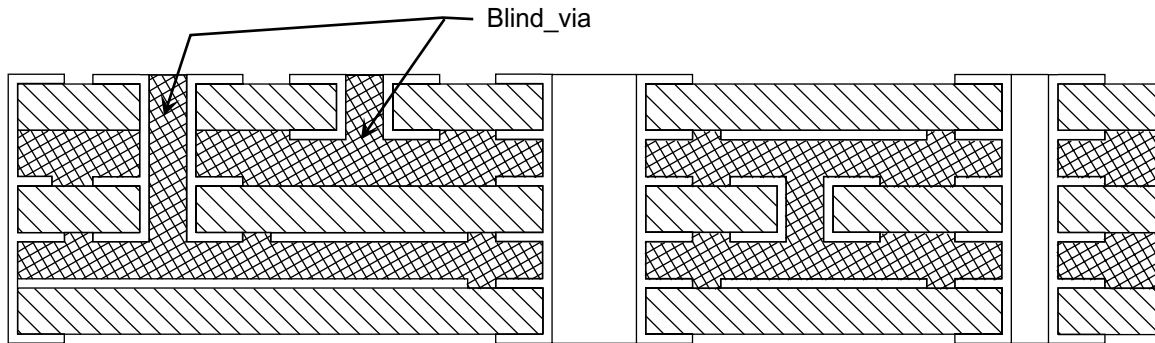


Figure 11 – Blind_via

Constraints:

WR1: Only one end of a Blind_via shall be at a surface of the Interconnect_module.

NOTE EXPRESS specification:

```

*)
ENTITY blind_via
  ABSTRACT SUPERTYPE OF (ONEOF (bonded_conductive_base_blind_via,
    non_conductive_base_blind_via, plated_conductive_base_blind_via))
  SUBTYPE OF (via);
END_ENTITY; -- blind_via

```

(*

4.2.72 Block

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Block is a type of Csg_primitive (see 4.2.201) which is used in forming a "CSG" solid model.

NOTE EXPRESS specification:

```

*)
ENTITY block
  SUBTYPE OF (csg_primitive);
END_ENTITY; -- block

( *
```

4.2.73 Bond_assembly_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bond_assembly_position is a type of Managed_design_object (see 4.2.524). A Bond_assembly_position is the placement of an Assembled_with_bonding (see 4.2.30) within the Physical_unit_3d_shape (see 4.2.666) of an assembly. The Bond_assembly_position is an approximation to the actual position since it uses a default shape for the bond. There may be considerable variation between the default shape and actual shapes in the "as delivered" product.

The data associated with a Bond_assembly_position are the following:

- assembly_model;
- bond;
- bond_definition_placement;
- bond_model.

4.2.73.1 assembly_model

The assembly_model specifies the Physical_unit_3d_shape (see 4.2.666) of the assembly for the Bond_assembly_position.

4.2.73.2 bond

The bond specifies the role of the Assembled_with_bonding (see 4.2.30) for the Bond_assembly_position.

4.2.73.3 bond_definition_placement

The bond_definition_placement specifies the role of the Axis_placement_3d (see 4.2.58) for the Bond_assembly_position. The placement specifies the transformation applied to the bond model to bring it into the assembly model.

4.2.73.4 bond_model

The bond_model specifies the Default_assembly_bond_shape (see 4.2.244) of the assembly bond for the Bond_assembly_position.

NOTE EXPRESS specification:

```

*)
ENTITY bond_assembly_position
  SUBTYPE OF (managed_design_object);
    bond : assembled_with_bonding;
    assembly_model : physical_unit_3d_shape;
    bond_model : default_assembly_bond_shape;
    bond_definition_placement : axis_placement_3d;
END_ENTITY; -- bond_assembly_position

( *
```

4.2.74 Bonded_conductive_base_blind_via

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bonded_conductive_base_blind_via is a type of Blind_via (see 4.2.71) in which the material that composes the base of the Via (see 4.2.947) is conductive, and in which the base material has been implemented by conductively bonding the Land (see 4.2.490) at the base of the Via to a Stratum_feature (see 4.2.851) of an adjacent conducting Stratum (see 4.2.845).

The data associated with a Bonded_conductive_base_blind_via are the following:

— features_join.

4.2.74.1 features_join

The features_join specifies the Stratum_feature_conductive_join (see 4.2.853) that joins the Land (see 4.2.490) on the interior facing surface of the Stratum (see 4.2.845) that is one end of the extent of the Blind_via (see 4.2.71) to the Land or other Stratum_feature (see 4.2.851) on the interior facing surface of the Stratum immediately adjacent to the first Stratum. The surfaces identified need be adjacent only through the extent of the intersection of these two joined features.

Constraints:

WR1: The Land at the base of the Bonded_conducted_base_blind_via shall be bonded to the Land on the adjacent Stratum.

NOTE A conductive bonding agent may be used to form the bond identified by features_join.

NOTE EXPRESS specification:

```

*)
ENTITY bonded_conductive_base_blind_via
  SUBTYPE OF (blind_via);
  features_join : stratum_feature_conductive_join;
END_ENTITY; -- bonded_conductive_base_blind_via

( *

```

4.2.75 Boolean_csg_result

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Boolean_csg_result is a type of Shape_item (see 4.2.818). A Boolean_csg_result is a Bound_volume_shape (see 4.2.76), or a Csg_primitive (see 4.2.201), which is the result of using a boolean operator. The Boolean_csg_result allows for the composition or decomposition of a Bound_volume_shape, or a Csg_primitive.

The data associated with a Boolean_csg_result are the following:

- csg_operator;
- operand_1;
- operand_2.

4.2.75.1 csg_operator

The csg_operator specifies the difference, the intersection, or the union that determines the Boolean_csg_result.

4.2.75.2 operand_1

The operand_1 specifies the Bound_volume_shape (see 4.2.76), or a Csg_primitive (see 4.2.201) which is the first operand for the Boolean_csg_result.

4.2.75.3 operand_2

The operand_2 specifies the Bound_volume_shape (see 4.2.76), or a Csg_primitive (see 4.2.201) which is the second operand for the Boolean_csg_result.

NOTE EXPRESS specification:

```

*)
ENTITY boolean_csg_result
  SUBTYPE OF (shape_item);
    operand_1      : boolean_csg_operand;
    operand_2      : boolean_csg_operand;
    csg_operator    : boolean_operator;
END_ENTITY; -- boolean_csg_result

--tt tc1 wd9 modified
( *
```

4.2.76 Bound_volume_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bound_volume_shape is a type of Shape_definition (see 4.2.808). A Bound_volume_shape is one of an Advanced_b_rep (see 4.2.6), a Csg_solid (see 4.2.202), an Edge_based_wireframe (see 4.2.310), a Geometrically_bounded_surface (see 4.2.425), a Shell_based_wireframe (see 4.2.820), or a Solid_of_linear_extrusion (see 4.2.833). A Bound_volume_shape is a three-dimensional geometric representation of a volume.

NOTE EXPRESS specification:

```

*)
ENTITY bound_volume_shape
  ABSTRACT SUPERTYPE OF (ONEOF (
    advanced_b_rep,
    csg_solid,
    edge_based_wireframe,
    geometrically_bounded_surface,
    shell_based_wireframe,
    solid_of_linear_extrusion))
  SUBTYPE OF (shape_definition);
END_ENTITY; -- bound_volume_shape

( *
```

4.2.77 Boundary_based_position_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Boundary_based_position_tolerance` is a type of `Position_tolerance_with_referenced_datum_system` (see 4.2.696) and a type of `Geometric_tolerance_with_applied_material_condition_property` (see 4.2.421). The `Boundary_based_position_tolerance` defines a `Tolerance_zone_boundary` (see 4.2.915) that is offset from either the least material condition boundary or the maximum material condition boundary depending on the value of the `applied_material_condition_property_name`.

NOTE The attribute `applied_material_condition_property_name` is declared in the `Geometric_tolerance_with_applied_material_condition_property` entity of which this entity is a subtype.

The least material condition boundary and the maximum material condition boundary are established either by one or more `Dimensional_characteristic` (see 4.2.298) or by a `Profile_of_any_surface_tolerance` (see 4.2.747). If the `Boundary_based_position_tolerance` specifies an `applied_material_condition_property_name` of "maximum_material_principle", the `Tolerance_zone_boundary` is offset from the maximum material condition boundary by half of the offset value and in the direction away from the least material condition boundary. If the `Boundary_based_position_tolerance` specifies an `applied_material_condition_property_name` of "least_material_requirement", then the `Tolerance_zone_boundary` is offset from the least material condition boundary by half of the offset value and in the direction away from the maximum material condition boundary.

NOTE Boundary based position tolerances are discussed in 5.10.1 and 6.5.5.1 of ASME Y14.5M.

The data associated with a `Boundary_based_position_tolerance` are the following:

— `profile_control`.

4.2.77.1 `profile_control`

The `profile_control` specifies a `Profile_of_any_surface_tolerance_with_referenced_datum_system` (see 4.2.748) that defines a `Tolerance_zone_boundary` (see 4.2.915) that establishes the least material condition boundary and the maximum material condition boundary from which the `Positional_boundary` (see 4.2.698) defined by the `Boundary_based_position_tolerance` is based. A `profile_control` need not be specified for a `Boundary_based_position_tolerance`.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` shall be a `Physical_feature_or_part_template`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

WR2: If a `Profile_of_any_surface_tolerance_with_referenced_datum_system` is specified as the `profile_control`, both the `Profile_of_any_surface_tolerance_with_referenced_datum_system` and the `Boundary_based_position_tolerance` shall specify the same instance of `Physical_feature_or_part_template` as their `toleranced_shape_element`.

WR3: If a `Profile_of_any_surface_tolerance_with_referenced_datum_system` is specified as the `profile_control`, the `Datum` in the `Datum_system` of the `Profile_of_any_surface_tolerance_with_referenced_datum_system` shall be in the `Datum_system` of the `Boundary_based_position_tolerance` and shall be assigned the same datum precedence.

WR4: If the `Boundary_based_position_tolerance` specifies a `Profile_of_any_surface_tolerance` as the `profile_control`, the `Boundary_based_position_tolerance` shall specify a `Profile_related_positional_boundary_definition` as the `tolerance_zone_or_boundary_definition`; otherwise, the `Boundary_based_position_tolerance` shall specify a `Positional_boundary_member_definition` as the `tolerance_zone_or_boundary_definition`.

NOTE EXPRESS specification:

```

*)
ENTITY boundary_based_position_tolerance
  SUBTYPE OF (position_tolerance_with_referenced_datum_system,
    geometric_tolerance_with_applied_material_condition_property);
  profile_control :
OPTIONAL profile_of_any_surface_tolerance_with_referenced_datum_system;
END_ENTITY; -- boundary_based_position_tolerance

( *
```

4.2.78 Boundary_size_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Boundary_size_characteristic` is a type of `Managed_design_object` (see 4.2.524). A `Boundary_size_characteristic` is either a `Diametrical_boundary_size_characteristic` (see 4.2.287) or an `Opposing_boundary_set_size_characteristic` (see 4.2.568). A `Boundary_size_characteristic` is a definition that characterizes the size of the `Tolerance_zone_boundary` (see 4.2.915).

The data associated with a `Boundary_size_characteristic` are the following:

- `boundary_definition`;
- `maximum_tolerance_value`;
- `tolerance_value`.

4.2.78.1 boundary_definition

The `boundary_definition` is an inverse relationship that specifies that the existence of the `Boundary_size_characteristic` is dependent on the existence of the `Boundary_zone_definition_with_specified_size` (see 4.2.79) that specifies the `Boundary_size_characteristic` as its `specified_boundary_size`.

4.2.78.2 maximum_tolerance_value

The maximum_tolerance_value specifies the Length_data_element (see 4.2.511) that represents the maximum size of the Tolerance_zone_boundary (see 4.2.915). The maximum_tolerance_value need not be specified for the Boundary_size_characteristic.

NOTE Fig. 6-42 of ASME Y14.5M illustrates the application of a maximum value to a geometric tolerance.

4.2.78.3 tolerance_value

The tolerance_value specifies the Length_data_element (see 4.2.511) that represents the size of the Tolerance_zone_boundary (see 4.2.915) for the Boundary_size_characteristic.

NOTE EXPRESS specification:

```

*)
ENTITY boundary_size_characteristic
  ABSTRACT SUPERTYPE OF (ONEOF (diametrical_boundary_size_characteristic,
    opposing_boundary_set_size_characteristic))
  SUBTYPE OF (managed_design_object);
    tolerance_value      : length_data_element;
    maximum_tolerance_value : OPTIONAL length_data_element;
  INVERSE
    boundary_definition : boundary_zone_definition_with_specified_size FOR
      specified_boundary_size;
END_ENTITY; -- boundary_size_characteristic

( *
```

4.2.79 Boundary_zone_definition_with_specified_size

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Boundary_zone_definition_with_specified_size is a type of Tolerance_zone_boundary_definition (see 4.2.916) in which the size of the Tolerance_zone_boundary (see 4.2.915) is defined by a Boundary_size_characteristic (see 4.2.78). A Boundary_zone_definition_with_specified_size may be a Profile_boundary_definition (see 4.2.742).

The data associated with a Boundary_zone_definition_with_specified_size are the following:

— specified_boundary_size.

4.2.79.1 specified_boundary_size

The specified_boundary_size specifies the Boundary_size_characteristic (see 4.2.78) that defines the size of the Tolerance_zone_boundary (see 4.2.915) for the Boundary_zone_definition_with_specified_size.

NOTE EXPRESS specification:

```

*)
ENTITY boundary_zone_definition_with_specified_size
  SUPERTYPE OF (profile_boundary_definition)
  SUBTYPE OF (tolerance_zone_boundary_definition);
  specified_boundary_size : boundary_size_characteristic;
END_ENTITY; -- boundary_zone_definition_with_specified_size

( *

```

4.2.80 Bounding_curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bounding_curve is a type of Curve . A Bounding_curve is provided in this part of ISO 10303 to describe the left side or the right side of a Conductive_interconnect_element_with_user_defined_single_transition (see 4.2.175) on an interconnect substrate.

EXAMPLE 1 Figure 12 illustrates a detail of a layout drawing that shows a transition on a trace. Figure 13 illustrate several alternative functions (only one of which is used in this example). A selected function describes the mapping from a constant to the shape of the top side of the trace segment identified as "I" in subfigure "C" in figure 12. The segment "I" is supported in this part of ISO 10303 through a member of Conductive_interconnect_element_with_user_defined_single_transition.

The data associated with a Bounding_curve are the following:

- basis_curve;
- mapping_function.

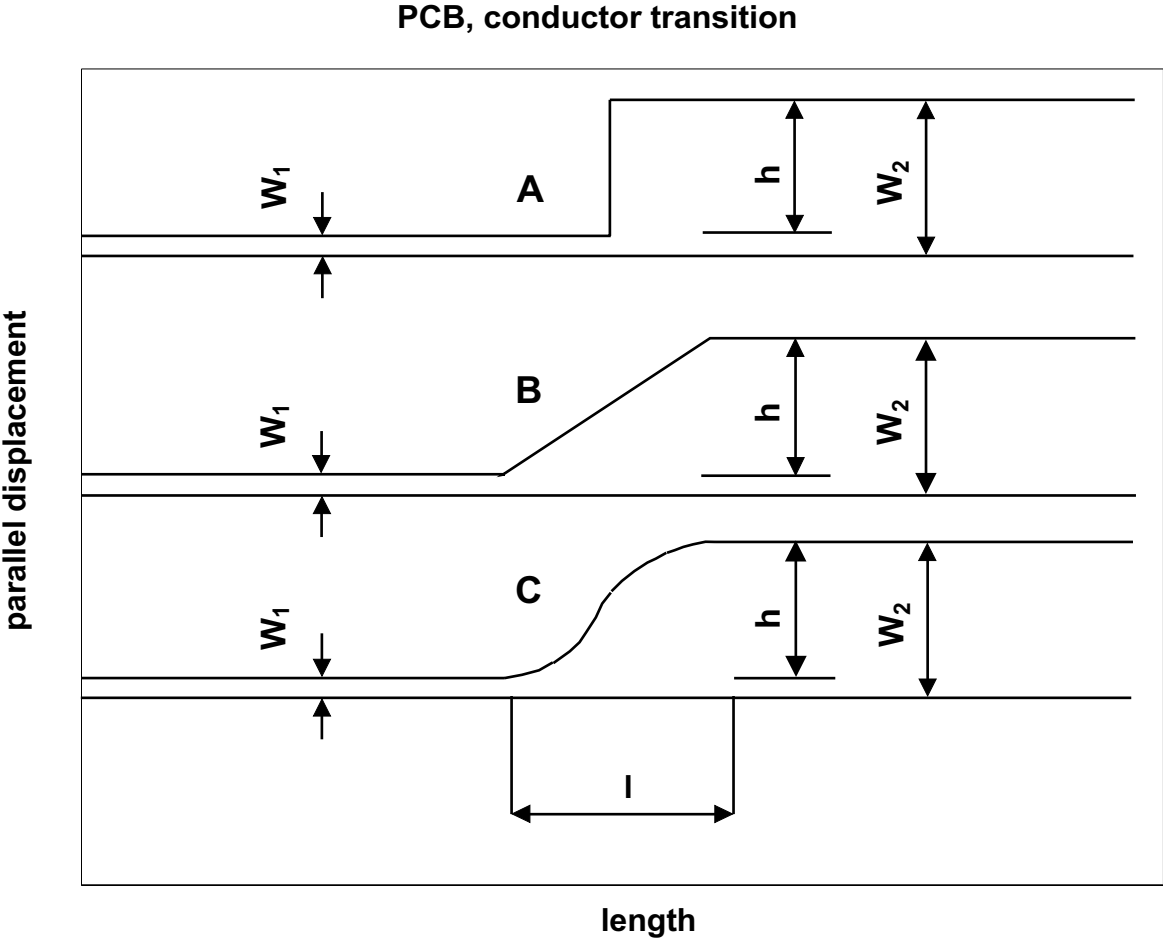


Figure 12 – Fabrication drawing detail

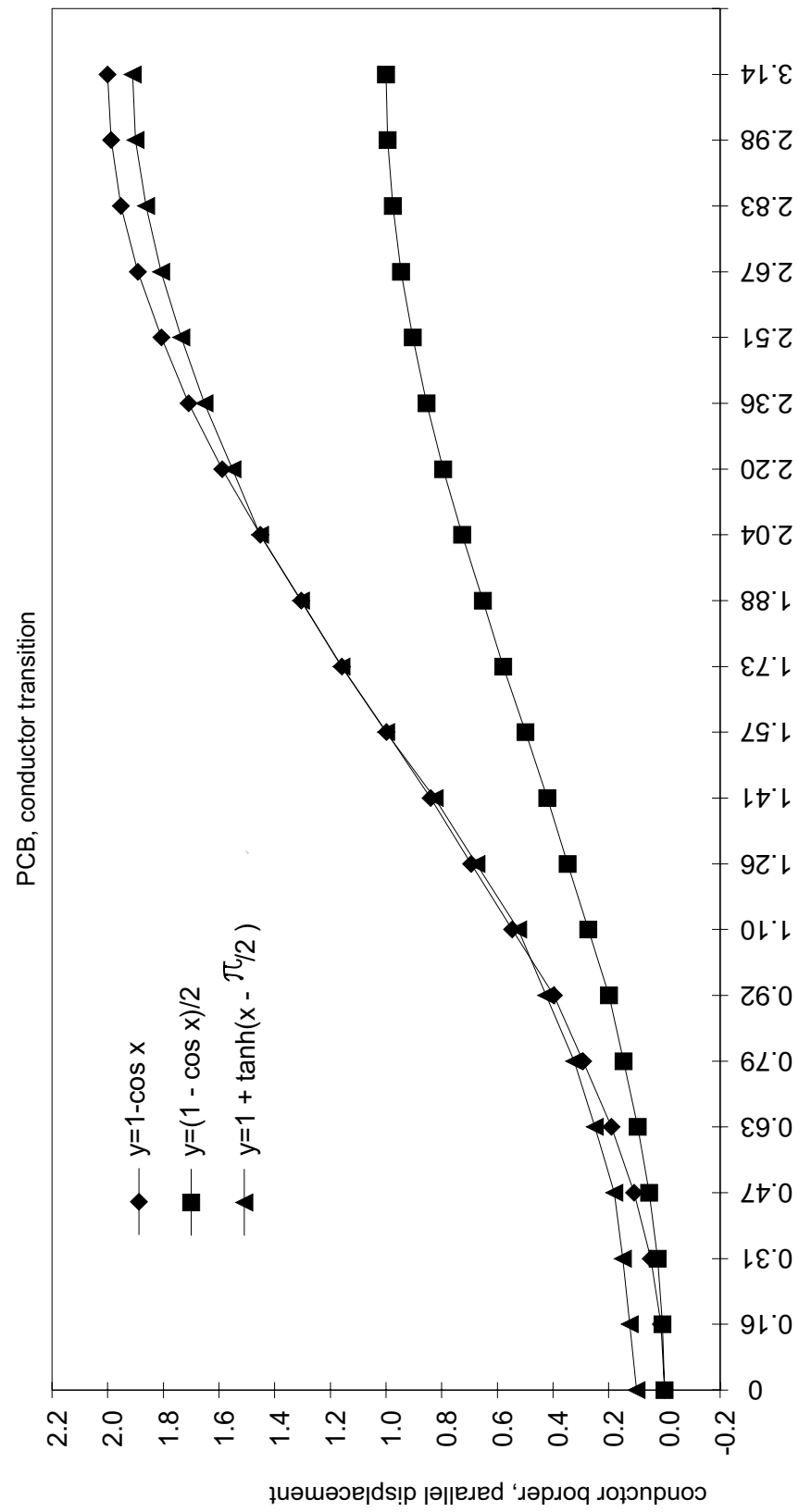


Figure 13 – Alternative functions for Bounding_Curve

4.2.80.1 basis_curve

The basis_curve specifies the role of the Trimmed_line (see 4.2.933) for the Bounding_curve. The basis_curve founds the Bounding_curve in the geometric context of the design.

4.2.80.2 mapping_function

The mapping_function specifies the role of the External_definition (see 4.2.361) for the Bounding_curve. The mapping_function maps the Trimmed_line onto the resultant curve which is the Bounding_curve.

NOTE This mechanism ensures that the geometry may be exchanged using this part of ISO 103030 even though the mapping function is an externally defined concept.

Constraints:

WR1: A Bounding_curve shall not specify itself through the basis_curve, either directly or indirectly.

WR2: The mapping_function shall have the property of being a one to one map.

NOTE EXPRESS specification:

```
*)
ENTITY bounding_curve
  SUBTYPE OF (curve);
    mapping_function : external_definition;
    basis_curve      : trimmed_line;
END_ENTITY; -- bounding_curve

( *
```

4.2.81 Buried_via

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Buried_via is a type of Via (see 4.2.947). Buried_via is defined in IEC 60050-541.

EXAMPLE 1 Figure 14 illustrates a Buried_via in a cross-section view of an interconnect substrate.

Constraints:

WR1: No end of the Buried_via shall be exposed in the Interconnect_module.

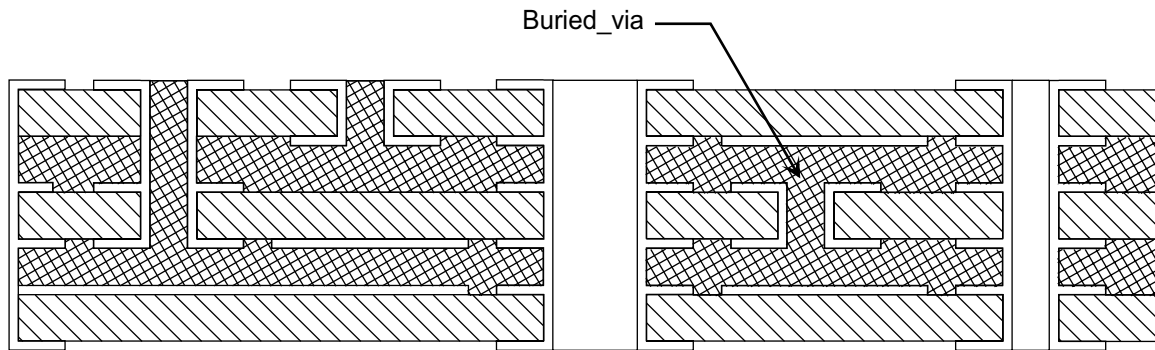


Figure 14 – Buried_via

NOTE EXPRESS specification:

```
*)
ENTITY buried_via
  SUBTYPE OF (via);
END_ENTITY; -- buried_via

--tt tc1 wd10 renamed
(*
```

4.2.82 Bus_element_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Bus_element_link is a type of Managed_design_object (see 4.2.524).

NOTE EXPRESS specification:

```
*)
ENTITY bus_element_link
  SUBTYPE OF (managed_design_object);
--tt tc1 wd2 change
  precedent_element      : bus_structural_definition_or_node;
  subsequent_element     : bus_structural_definition_or_node;
  UNIQUE
  UR1 : precedent_element, subsequent_element;
  WHERE
  WR1: precedent_element &#60;&#62; subsequent_element;
END_ENTITY; -- bus_element_link
(*
```

4.2.83 Bus_structural_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Bus_structural_definition` is a type of `Managed_design_object` (see 4.2.524). A `Bus_structural_definition` is the aggregate or composition of the mathematical or logical structure of the bus. If a bus were to be represented by a graph, the `Bus_structural_definition` would be a root node for the bus in that representation.

NOTE 1 A bus is a composition concept. Once defined, the bus definition can be used in various connectivity related aspects of the design. Several of the possible usages are conflicting and rules are provided to ensure unambiguous data. The structure of the reference model prohibits certain conflicting usages from occurring simultaneously.

NOTE 2 Ordered definition of buses is supported in this part of ISO 10303 by interpreting the name of the bus element as indicating the position in the sequence, using normal ASCII ordering for regular expressions. Spaces are not allowed in names.

The data associated with a `Bus_structural_definition` are the following:

- `bus_name`;
- `composition`;
- `external_bus_access`.

4.2.83.1 `bus_name`

The `bus_name` specifies the string identifying the `Bus_structural_definition`.

NOTE A project may establish project wide default names for buses. The identifier should then be globally unique within that project.

4.2.83.2 `composition`

The composition is an inverse relationship that specifies that the existence of the `Bus_structural_definition` is dependent on the existence of the `Bus_element_relationship` (see ??) that specifies the `Bus_structural_definition` as its `bus_definition`.

4.2.83.3 `external_bus_access`

The `external_bus_access` is an inverse relationship that specifies that there shall be no more than one `Functional_unit_network_terminal_definition_bus_assignment` (see 4.2.404) that specifies the `Bus_structural_definition` as its `connected_bus`. The `external_bus_access` need not be supplied for a particular `Bus_structural_definition`.

Constraints:

UR1: The bus_name shall be unique within a population of Bus_structural_definitions.

NOTE EXPRESS specification:

```

*)
ENTITY bus_structural_definition
  SUBTYPE OF (managed_design_object);
  bus_name : ee_name;
--tt tc1 wd10 modified
  composition : SET [1:?] OF bus_element_link;
  INVERSE
    external_bus_access :
  SET [0:1] OF functional_unit_network_terminal_definition_bus_assignment FOR
  connected_bus;
  UNIQUE
    UR1 : bus_name;
--tt tc1 wd13 added
  WHERE
    WR1: consistent_bus_structural_definition(composition);
END_ENTITY; -- bus_structural_definition

--tt tc1 wd9 modified
(

```

4.2.84 Cable_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cable_component is a

The data associated with a Cable_component are the following:

— additional_data.

4.2.84.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY cable_component
  SUBTYPE OF (physical_component);
END_ENTITY; -- routed_physical_component

( *
```

4.2.85 Capacitance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Capacitance_data_element is a

The data associated with a Capacitance_data_element are the following:

— additional_data.

4.2.85.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY capacitance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- capacitance_data_element

( *
```

4.2.86 Cartesian_coordinate_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cartesian_coordinate_system is a type of Managed_design_object (see 4.2.524). A Cartesian_coordinate_system is a system of two coordinates that locates a point on a plane and measures its distance from either of two intersecting straight-line axes along a line parallel to the other axis, or a system of three coordinates that locates a point in space and measures its distance from any of three intersecting coordinate planes measured parallel to that one of three straight-line axes that is the intersection of the other two planes.

The data associated with a Cartesian_coordinate_system are the following:

- angle_unit_of_measure;
- angular_uncertainty;
- dimensionality;
- length_unit_of_measure;
- length_uncertainty.

4.2.86.1 angle_unit_of_measure

The angle_unit_of_measure specifies either degree or radian as the measurement unit of the Cartesian_coordinate_system.

4.2.86.2 angular_uncertainty

The angular_uncertainty specifies the Angle_data_element (see 4.2.23) of the uncertainty of angularity for the Cartesian_coordinate_system.

NOTE This value is based on the precision of the processor that created the model.

4.2.86.3 dimensionality

The dimensionality specifies either the two-dimensional or three-dimensional nature of the Cartesian_coordinate_system.

4.2.86.4 length_unit_of_measure

The length_unit_of_measure specifies the string for the unit of measurement for the Cartesian_coordinate_system. This shall be interpreted as a length unit of measure.

4.2.86.5 length_uncertainty

The length_uncertainty specifies the Length_data_element (see 4.2.511) of the uncertainty of length for the Cartesian_coordinate_system.

NOTE This value is based on the precision of the processor that created the model.

NOTE EXPRESS specification:

```

*)
ENTITY cartesian_coordinate_system
  SUBTYPE OF (managed_design_object);
    length_unit          : STRING;
    length_unit_prefix   : OPTIONAL si_prefix;
    length_dynamic_range : OPTIONAL INTEGER;
    dimensionality       : two_or_three_dimensional;
    angle_unit           : degree_or_radian;
    length_uncertainty   : length_data_element;
    angular_uncertainty  : OPTIONAL angle_data_element;
END_ENTITY; -- cartesian_coordinate_system
( *
```

4.2.87 Cartesian_point

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cartesian_point is a type of Shape_item (see 4.2.818). A Cartesian_point is a geometric point in cartesian space. The requirements of the domain are for two-dimensional and three-dimensional geometry. Each usage of geometry in the domain will specify the required dimensionality.

NOTE EXPRESS specification:

```

*)
ENTITY cartesian_point
  SUBTYPE OF (shape_item);
END_ENTITY; -- cartesian_point
( *
```

4.2.88 Cartesian_transformation_operator_2d

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Cartesian_transformation_operator_2d` is a type of `Managed_design_object` (see 4.2.524). A `Cartesian_transformation_operator_2d` defines a geometric transformation in two-dimensional space composed of translation, rotation, mirroring and uniform scaling. A precise mathematical definition for `Cartesian_transformation_operator_2d` is provided in ISO 10303-42.

The data associated with a `Cartesian_transformation_operator_2d` are the following:

- `axis_1`;
- `axis_2`;
- `mirror`;
- `scale`;
- `translation`.

4.2.88.1 axis_1

The `axis_1` specifies one role of a `Direction` (see 4.2.304) for the `Cartesian_transformation_operator_2d`.

4.2.88.2 axis_2

The `axis_2` specifies one role of a `Direction` (see 4.2.304) for the `Cartesian_transformation_operator_2d`.

4.2.88.3 mirror

The `mirror` is a derived attribute that specifies the value of the determinant of the transformation matrix for the `Cartesian_transformation_operator_2d`.

4.2.88.4 scale

The `scale` specifies the role of the `Positive_ratio_measure` (see ??) for the `Cartesian_transformation_operator_2d`.

4.2.88.5 translation

The `translation` specifies the role of the `Cartesian_point` (see 4.2.87) for the `Cartesian_transformation_operator_2d`.

Constraints:

IP1: Mirroring shall be only about the local Y axis, which "local Y axis" is defined as a): being parallel to the Y axis of the current coordinate system and b): intersecting the Cartesian_point which is in the role of translation for this Cartesian_transformation_operator_2d.

EXAMPLE 1 A point at (X = 3, Y = 2) would be at (X = -3, Y = 2) after being transformed using the Cartesian_transformation_operator_2d if no rotation were applied and no translation were applied, but mirroring were applied.

NOTE Typically the transformation matrix is the result of applying the ordered sequence of "rotate, mirror, translate" operations to an identity matrix. Placing translate last in the sequence satisfies the requirement that mirroring shall be about a local axis.

IP2: The determinant of the transformation matrix shall have a value of -1.0 or +1.0.

NOTE In a received file the value of the determinant may not be exactly 1.0 or -1.0 due to roundoff error. Since these are the only two allowed values, standard techniques to deal with the error are easily applied.

IP3: The scale value for the transformation shall be one.

NOTE EXPRESS specification:

```

*)
ENTITY cartesian_transformation_operator_2d
  SUBTYPE OF (managed_design_object);
    axis_1          : direction;
    axis_2          : direction;
    translation     : cartesian_point;
    scale           : positive_ratio_measure;
DERIVE
  mirror : BOOLEAN := derive_mirror(SELF);
END_ENTITY; -- cartesian_transformation_operator_2d
( *

```

4.2.89 Celsius_temperature_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Celsius_temperature_data_element is a

The data associated with a Celsius_temperature_data_element are the following:

— additional_data.

4.2.89.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY celsius_temperature_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- celsius_temperature_data_element

( *
```

4.2.90 Centre_axis

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Centre_axis is a type of Centre_of_symmetry (see 4.2.91) that is axial in form. The Shape_element (see 4.2.810) from which the Centre_axis is derived are associated with the Centre_axis by the Shape_element_deriving_relationship (see 4.2.814) that are specified as the deriving_relationships.

NOTE 1 The attribute deriving_relationships is declared in the Derived_shape_element (see 4.2.260) entity of which this entity is a subtype.

NOTE 2 The precise interpretation of a Centre_axis is context dependent. That is, the distinction is not made between an axis of a feature and a derived median line of a feature as defined in 1.3 of ASME Y14.5M.

EXAMPLE 1 A Position_tolerance (see 4.2.695) and a Straightness_tolerance (see 4.2.844) may be applied to the same Centre_axis of a cylindrical Shape_element.

Constraints:

IP1: The form of the Shape_elements from which the Centre_axis is derived shall be consistent with the form of shape elements from which a centre axis can be derived.

NOTE EXPRESS specification:

```
*)
ENTITY centre_axis
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_axis

( *
```

4.2.91 Centre_of_symmetry

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Centre_of_symmetry is a type of Derived_shape_element (see 4.2.260) that is derived by determining the geometric centre of one or more Shape_element (see 4.2.810). A Centre_of_symmetry is one of a Centre_axis (see 4.2.90), a Centre_plane (see 4.2.92), or a Centre_point (see 4.2.93).

NOTE EXPRESS specification:

```
*)
ENTITY centre_of_symmetry
  ABSTRACT SUPERTYPE OF (ONEOF (centre_axis,centre_plane,centre_point))
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- centre_of_symmetry
```

(*

4.2.92 Centre_plane

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Centre_plane is a type of Centre_of_symmetry (see 4.2.91) that is planar in form. The Shape_element (see 4.2.810) derived from the Centre_plane is associated with the Centre_plane by the Shape_element_deriving_relationship (see 4.2.814) that are specified as the deriving_relationships.

NOTE 1 The attribute deriving_relationships is declared in the Derived_shape_element entity of which this entity is a subtype.

NOTE 2 The precise interpretation of a Centre_plane is context dependent. That is, the distinction is not made between the centre plane of a feature and a derived median plane of a feature as defined in 1.3 of ASME Y14.5M.

EXAMPLE 1 A Position_tolerance and a Straightness_tolerance may be applied to a Centre_plane that is derived from a Composite_unit_shape_element that consists of two opposing surfaces of a notch (or tab).

Constraints:

IP1: The form of the Shape_elements from which the Centre_plane is derived shall be consistent with the form of shape elements from which a centre plane can be derived.

NOTE EXPRESS specification:

```
*)
ENTITY centre_plane
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_plane
```

(*

4.2.93 Centre_point

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Centre_point is a type of Centre_of_symmetry (see 4.2.91) the form of which is that of a point. The Shape_element (see 4.2.810) from which the Centre_point is derived are associated with the Centre_point by the Shape_element_deriving_relationship (see 4.2.814) that are specified as the deriving_relationships.

NOTE The attribute deriving_relationships is declared in the Derived_shape_element (see 4.2.260) entity of which this entity is a subtype.

Constraints:

IP1: The form of the Shape_elements from which the Centre_point is derived shall be consistent with the form of shape elements from which a centre point can be derived.

NOTE EXPRESS specification:

```
*)
ENTITY centre_point
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_point

( *
```

4.2.94 Certification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Certification is a type of Managed_design_object (see 4.2.524). A Certification is a granting of approval for use of a Supplied_product_version (see 4.2.879).

The data associated with a Certification are the following:

- certification_approval;
- certification_date;
- certification_number;
- certified_product_version.

4.2.94.1 certification_approval

The certification_approval specifies the Ee_approval (see 4.2.314) of the Certification.

4.2.94.2 certification_date

The certification_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) for the Certification.

4.2.94.3 certification_number

The certification_number specifies the Ee_text (see 4.2.339) that identifies the Certification.

4.2.94.4 certified_product_version

The certified_product_version specifies the Supplied_product_version (see 4.2.879) for the subject of the Certification.

Constraints:

UR1: The certification_number shall be unique within a population of Certifications.

NOTE EXPRESS specification:

```

*)
ENTITY certification
  SUBTYPE OF (managed_design_object);
    certification_approval    : ee_approval;
    certification_number      : ee_text;
    certified_product_version : supplied_product_version;
    certification_date        : date_or_date_and_time;
  UNIQUE
    UR1 : certification_number;
END_ENTITY; -- certification

( *
```

4.2.95 Change_design_object_management_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Change_design_object_management_relationship` is a type of `Design_object_management_relationship` (see 4.2.278) that provides information about design changes, including the swapping changes among components, gates, or terminals. A `Change_design_object_management_relationship` may be a `Component_swap_specification` (see 4.2.136), a `Gate_path_swap_specification` (see 4.2.415), or a `Terminal_swap_specification` (see 4.2.891).

The data associated with a `Change_design_object_management_relationship` are the following:

- `current_design_object`;
- `previous_design_object`.

4.2.95.1 `current_design_object`

The `current_design_object` specifies the `Managed_design_object` (see 4.2.524), after the design change, for the `Change_design_object_management_relationship`. The `current_design_object` is in the scope of the `current_design`.

4.2.95.2 `previous_design_object`

The `previous_design_object` specifies the `Managed_design_object` (see 4.2.524), before the design change, for the `Change_design_object_management_relationship`. The `previous_design_object` is in the scope of the `previous_design`.

Constraints:

UR1: The combination of the `current_design_object` and the `current_design` inherited from `Design_object_management_relationship` shall be unique within a population of `Change_design_object_management_relationships`.

UR2: The combination of the `previous_design_object` and the `previous_design` inherited from `Design_object_management_relationship` shall be unique within a population of `Change_design_object_management_relationships`.

WR1: The `current_design_object` and the `previous_design_object` shall be of the same type of application object, as defined in this part of ISO 10303.

NOTE A design change may affect only the attributes of the object and not its identifier.

NOTE EXPRESS specification:

```

*)
ENTITY change_design_object_management_relationship
  SUPERTYPE OF (ONEOF (terminal_swap_specification,
    gate_path_swap_specification,
    component_swap_specification))
  SUBTYPE OF (design_object_management_relationship);
  current_design_object : managed_design_object;
  previous_design_object : managed_design_object;
  UNIQUE
    UR1 : current_design_object, current_design;
    UR2 : previous_design_object, previous_design;
END_ENTITY; -- change_design_object_management_relationship

( *

```

4.2.96 Change_order

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Change_order is a type of Work_order (see 4.2.964) that authorizes the development of a modified design of an Ee_product (see 4.2.319), resulting in a new Ee_product_version (see 4.2.328).

The data associated with a Change_order are the following:

— adopted_solution.

4.2.96.1 adopted_solution

The adopted_solution specifies the Design_object_management_relationship (see 4.2.278), or Ee_text (see 4.2.339) which is to be implemented by the Change_order.

NOTE EXPRESS specification:

```

*)
ENTITY change_order
  SUBTYPE OF (work_order);
  adopted_solution : design_object_management_relationship_or_description;
END_ENTITY; -- change_order

( *

```

4.2.97 Change_request

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Change_request is a type of Work_request (see 4.2.965) that identifies the proposed extent of work to be done to modify the design of a product.

The data associated with a Change_request are the following:

- consequence;
- recommended_solution;
- version.

4.2.97.1 consequence

The consequence specifies the Ee_text (see 4.2.339) that describes the impact that the Change_request will have on the product. There shall be one or more Ee_text for a Change_request.

4.2.97.2 recommended_solution

The recommended_solution specifies one or more Design_object_management_relationship (see 4.2.278), or Ee_text (see 4.2.339) of a potential solution which meets the requirements defined in the Change_request. A recommended_solution need not be specified for a Change_request.

4.2.97.3 version

The version specifies the string identifying a Change_request. The version shall be human interpretable.

Constraints:

UR1: The combination of the version and the work_request_id inherited from Work_request shall be unique within a population of Change_request.

NOTE EXPRESS specification:

```
*)
ENTITY change_request
  SUBTYPE OF (work_request);
    recommended_solution : OPTIONAL SET [1:?] OF
                                design_object_management_relationship_or_description;
    consequence          : SET [1:?] OF ee_text;
    version              : ee_name;
  UNIQUE
    UR1 : version, SELF\work_request.work_request_id;
END_ENTITY; -- change_request

( *
```

4.2.98 Characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Characteristic is a type of Managed_design_object (see 4.2.524). A Characteristic may be one of a Compound_characteristic (see 4.2.154), or a Numeric_parameter (see 4.2.565). A Characteristic is a property of an item.

NOTE A Characteristic may describe either a physical item, like a realized part, or a description that is unrealizable, such as a simulation model.

The data associated with a Characteristic are the following:

- property_value;
- reference_document;
- type_name.

4.2.98.1 property_value

The property_value specifies either a Data_element (see 4.2.211), an External_definition (see 4.2.361), or one of a boolean type, a LOGICAL, or a string type, for the Characteristic.

4.2.98.2 reference_document

The reference_document specifies the Ee_document (see 4.2.316) or a string identifying this Characteristic.

NOTE This part of ISO 10303 explicitly supports the unambiguous exchange of properties whose names are registered by a registration authority and whose definitions are included in data dictionaries. This part of ISO 10303 includes facilities to unambiguously exchange a wide variety of units including SI, user named, and derived.

4.2.98.3 type_name

The type_name specifies the string for this Characteristic.

Constraints:

UR1: The string representing the type_name shall not appear more than once in the reference_document in the role of "type name for Characteristic".

NOTE EXPRESS specification:

```
*)
ENTITY characteristic
  SUPERTYPE OF (ONEOF(compound_characteristic,
    numeric_parameter))
  SUBTYPE OF (managed_design_object);
  type_name          : STRING;
  reference_document : ee_document;
  property_value     : numeric_or_non_numeric_select;
END_ENTITY; -- characteristic
(*
```

4.2.99 Characteristic_data_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Characteristic_data_set is a type of Compound_characteristic (see ??). A Characteristic_data_set is a part of the data needed to compose a data table. The items included in the set are related by being in the set but there is no sequence specified. Specific functional relationships are beyond the scope of this Application object and should be addressed by the Analytical_model Application object.

NOTE EXPRESS specification:

```
*)
ENTITY characteristic_data_set
  SUBTYPE OF (compound_characteristic);
  SELF\characteristic.property_value : set_characteristic;
END_ENTITY; -- characteristic_data_set
(*
```

4.2.100 Characteristic_data_table

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Characteristic_data_table is a type of Compound_characteristic (see 4.2.154). A Characteristic_data_table describes data as a sequence of columns, where the elements in each successive column in the table are correlated according to the position of the column in the sequence of columns. The recursive construction shall terminate in a simple Characteristic Application object.

NOTE EXPRESS specification:

```

*)
ENTITY characteristic_data_table
  SUBTYPE OF (compound_characteristic);
  SELF\characteristic.property_value : list_characteristic;
END_ENTITY; -- characteristic_data_table
( *

```

4.2.101 Circular_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Circular_area is a type of Primitive_2d (see 4.2.720).

The data associated with a Circular_area are the following:

- centre;
- radius.

4.2.101.1 centre

The centre specifies a Cartesian_point (see 4.2.87) for the Circular_area.

4.2.101.2 radius

The radius specifies a Length_data_element (see 4.2.511) for the Circular_area.

NOTE EXPRESS specification:

```

*)
ENTITY circular_area
  SUBTYPE OF (primitive_2d);
  centre: cartesian_point;
  radius: length_data_element;
END_ENTITY; -- circular_area
( *

```

4.2.102 Circular_runout_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Circular_runout_tolerance` is a type of `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422) that specifies or partially specifies the design requirement on the run-out of circular elements of a `Physical_feature_or_part_template` (see 4.2.658) in relation to a `Datum_axis` (see 4.2.215). Where run-out is the quality or state of having perfect form and axial alignment in relation to an axis. The `Physical_feature_or_part_template` may be coaxial or normal to the `Datum_axis`. If the `Circular_runout_tolerance` is applied at a specified angle the defined `Tolerance_zone_definition` (see 4.2.920) is a `Runout_tolerance_zone_definition_with_specified_angle` (see 4.2.796).

NOTE 1 A circular run-out tolerance is typically verified by placing a full movement indicator against the surface of the part at the desired orientation and location and then rotating the part one complete revolution about the datum axis. The value of the full indicator movement is then read. This process is repeated at each of the desired test locations. If any one of the full indicator movement readings is greater than the specified tolerance value the surface has exceeded one of the tolerance zones and the part fails the inspection. In theory, there are an infinite number of test locations. However, in practice, a finite number of test locations are measured.

NOTE 2 Circular run-out tolerance is defined in terms of tolerance zone definitions in 14.13 of ISO 1101.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` shall be a `Physical_feature_or_part_template`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

WR2: The `Circular_runout_tolerance` shall not be a `Geometric_tolerance_with_applied_material_condition_property`.

WR3: There shall be at most two `Datum` in the `Datum_system` the definition of which is specified as the `datum_system_definition`.

NOTE The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` entity of which this entity is a subtype.

WR4: There shall be a `Datum_axis` in the `Datum_system` the definition of which is specified as the `datum_system_definition`.

WR5: The `Datum_system_defining_conditions` that is specified as the `datum_system_definition` shall be a `Datum_system_defining_conditions_without_material_conditions`.

WR6: The `Circular_runout_tolerance` shall specify a `Tolerance_zone_definition` as its `tolerance_zone_or_boundary_definition`; and that `Tolerance_zone_definition` shall specify a `Tolerance_zone_boundary_definition_with_specified_size` as its `boundary_definition`; and that `Tolerance_zone_boundary_definition_with_specified_size` shall specify an `Opposing_boundary_set_size_characteristic` as its `specified_boundary_size`.

IP1: The form of the toleranced `Physical_feature_or_part_template` shall be consistent with the form of a physical feature for which the application of a circular run-out tolerance is sensible.

NOTE EXPRESS specification:

```

*)
ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- circular_runout_tolerance

( *
```

4.2.103 Circularity_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Circularity_tolerance is a type of Geometric_tolerance_without_referenced_datum_system (see 4.2.423) that specifies or partially specifies the design requirement on the circularity, or roundness, of either a Shape_element (see 4.2.810)(if the Shape_element specified as the tolerated_shape_element is circular, essentially two-dimensional); or of the cross-sectional elements of the tolerated Shape_element (if it is either spherical or can be described as a surface of revolution). For a spherical Shape_element, the cross-sectional elements pass through the centre of the sphere. For a surface of revolution, the cross-sectional elements are taken normal to the axis of revolution. Each circular element must lie within its tolerance zone. The Tolerance_zone_boundary (see 4.2.915) consists of two concentric, coplanar circles. For a surface of revolution or spherical Shape_element the Circularity_tolerance defines a Tolerance_zone (see 4.2.913) that is representative of the tolerance zone for each of the cross-sectional elements.

NOTE 1 Circularity_tolerance is defined in terms of tolerance zone definitions in 14.3 of ISO 1101.

NOTE 2 The attribute tolerated_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

Constraints:

WR1: The Shape_element specified by the tolerated_shape_element shall be a Physical_feature_or_part_template.

WR2: The Circularity_tolerance shall not be a Geometric_tolerance_with_applied_material_condition_property.

WR3: The Circularity_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

IP1: The form of the tolerated Physical_feature_or_part_template shall be consistent with the form of a feature for which the application of a circularity tolerance is sensible.

NOTE EXPRESS specification:

```
*)  
ENTITY circularity_tolerance  
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);  
END_ENTITY; -- circularity_tolerance  
  
(*
```

4.2.104 Closed_curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Closed_curve is a type of Curve (see 4.2.203) that is traced by a moving point that returns to an arbitrary starting Cartesian_point (see 4.2.87).

Constraints:

IP1: A Closed_curve shall not be self intersecting.

NOTE EXPRESS specification:

```
*)  
ENTITY closed_curve  
  SUBTYPE OF (curve);  
END_ENTITY; -- closed_curve  
  
(*
```

4.2.105 Common_datum

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Common_datum is a type of Datum (see 4.2.214) that is established by two or more Datum_feature (see 4.2.226).

NOTE Common datums are described in 6.2.2 of ISO 5459.

Constraints:

WR1: Each Common_datum shall participate in establishing at least one Datum_system.

WR2: Each Common_datum shall be specified as the relating_shape_element by two or more Datum_feature_usage_in_common_datum_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY common_datum
  SUBTYPE OF (datum);
END_ENTITY; -- common_datum

( *
```

4.2.106 Component_2d_edge_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_2d_edge_location is a type of Component_2d_location (see 4.2.108). The Component_2d_edge_location places components on the edge of a substrate in order to maintain contact with both the top and bottom circuitry.

NOTE Connection_zone_interface_plane_relationship (see 4.2.188) and Connection_zone_based_assembly_joint (see 4.2.186) may be used to provide more details if needed by a receiving organization.

The data associated with a Component_2d_edge_location are the following:

— reference_terminal_assembly_joint.

4.2.106.1 reference_terminal_assembly_joint

The reference_terminal_assembly_joint specifies the Assembly_joint (see 4.2.41) that joins the Primary_reference_terminal (see 4.2.718) of the component being located to the associated mating component feature. The surface the mating component feature is associated with is part of the orientation description for the Component_2d_edge_location.

Constraints:

WR1: stuff to be added

NOTE EXPRESS specification:

```

*)
ENTITY component_2d_edge_location
  SUBTYPE OF (component_2d_location);
    reference_terminal_assembly_joint : assembly_joint;
--tt tc1 wd13 added
    mounting_surface_assembly_joint : assembly_joint;
--tt tc1 wd13 added
WHERE
  WR1: reference_terminal_assembly_joint &#60;&#62; mounting_surface_assembly_joint;
  WR2: SELF\component_2d_location.substrate_location = FALSE;
  WR3: SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
    'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE']) *
    TYPEOF(mounting_surface_assembly_joint.assembly_features_1)) &#62;= 0;
END_ENTITY; -- component_2d_edge_location

--tt tc1 wd8 modified
(

```

4.2.107 Component_2d_embedded_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_2d_embedded_location is a type of Component_2d_location (see 4.2.108). The Component_2d_embedded_location places components in a substrate under the constraint that is is required to maintain contact with Stratum_surface on both the top and bottom of the component.

NOTE The Embedded_component_terminal referenced by the Fabrication_joint referenced by reference_joint should be a constituent of one of the surfaces referenced by precedent_surface or subsequent_surface.

The data associated with a Component_2d_embedded_location are the following:

- precedent_surface;
- subsequent_surface;
- reference_terminal.

4.2.107.1 precedent_surface

The precedent_surface specifies the Adjacent_stratum_surface_embedded_component_surface_definition (see 4.2.4) that joins the Component_feature (see 4.2.118) of the component being located to the associated stratum surface. The Adjacent_stratum_surface_embedded_component_surface_definition in this role is identifying the stratum surface that is closer to the primary stratum in the substrate than is the embedded component itself.

4.2.107.2 subsequent_surface

The subsequent_surface specifies the Adjacent_stratum_surface_embedded_component_surface_definition (see 4.2.4) that joins the Component_feature (see 4.2.118) of the component being located to the associated stratum surface. The Adjacent_stratum_surface_embedded_component_surface_definition in this role is identifying the stratum surface that is further away from the primary stratum in the substrate than is the embedded component itself.

4.2.107.3 reference_terminal

The reference_terminal specifies the Fabrication_joint (see 4.2.368) that joins the Primary_reference_terminal (see 4.2.718) of the component being located to the Land_join_terminal or Non_functional_land_join_terminal that implements the internal connection to the circuitry in the rest of the substrate.

Constraints:

WR1: The Embedded_component_terminal referenced by reference_joint shall reference the same component as that referenced by the Component_feature referenced by the Adjacent_stratum_surface_embedded_component_surface_definition referenced by either the precedent_surface or the subsequent_surface.

NOTE EXPRESS specification:

```

*)
ENTITY component_2d_embedded_location
  SUBTYPE OF (component_2d_location);
    precedent_surface      : adjacent_stratum_surface_embedded_component_surface_definition;
    subsequent_surface     : adjacent_stratum_surface_embedded_component_surface_definition;
    reference_terminal     : fabrication_joint;
UNIQUE
  UR1 : precedent_surface, reference_terminal, subsequent_surface;
WHERE
  WR1: precedent_surface &#60;&#62; subsequent_surface;
  WR2: SELF\component_2d_location.substrate_location = FALSE;
END_ENTITY; -- component_2d_embedded_location
( *
```

4.2.108 Component_2d_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_2d_location` is a type of `Managed_design_object` (see 4.2.524). A `Component_2d_location` may be a `Component_2d_edge_location` (see 4.2.106). The `Component_2d_location` is the placement location, including orientation and mirroring, specified by a combination of description characteristics, substrate surface identification, and two-dimensional transformation. This Application Object requires `Component_assembly_2d_position` (see 4.2.112) for completeness. Mirroring shall not be applied to the location of a base substrate. Mirroring applied to components other than a base substrate shall indicate that the components are mounted on the surface of the substrate opposite to the primary surface. When this application object is used in this manner, there shall be only one base substrate in the assembly.

NOTE Subtypes of this application object includes sufficient information to accurately determine machine parameters for reliable installation of a component on an interconnect substrate. The use of this application object should be reserved for locating base substrates that are not themselves mounted in the design, or where the only mating condition information is encoded in the transformation.

The data associated with a `Component_2d_location` are the following:

- `placement_fixed`;
- `transformation`.

4.2.108.1 `placement_fixed`

The `placement_fixed` specifies whether or not the `Component_2d_location` is fixed in the design. A value of `TRUE` indicates that the `Component_2d_location` is fixed; a value of `FALSE` indicates that the `Component_2d_location` is not fixed.

4.2.108.2 `transformation`

The transformation specifies the role of the `Cartesian_transformation_operator_2d` (see 4.2.88) for the `Component_2d_location`.

Constraints:

IP1: The transformation matrix value of +1.0 shall indicate primary surface mounted component (un-mirrored). The transformation matrix value of -1.0 shall indicate secondary surface mounted component (mirrored). A mirror requirement to mirror about the local Y axis indicates that values on the Y axis are invariant, while values on the X axis are assigned the negative of the original value.

EXAMPLE 1 A point at (X = 3, Y = 2) would be at (X = -3, Y = 2) after being mirrored.

NOTE This standard does not support a "mounting style" attribute. Components which are altered from their normal mounting arrangement (e.g., "upside down mounting style") are considered to be `Altered_packaged_part` (see 4.2.10) in the context of this standard if the normal mounting arrangement is available and a relationship to that normal mounting arrangement is desired.

IP2: The scale value for the transformation shall be one.

NOTE EXPRESS specification:

```

*)
ENTITY component_2d_location
  SUPERTYPE OF (ONEOF(component_2d_edge_location,
                        component_2d_embedded_location,
                        component_2d_stacked_location,
                        component_2d_surface_location))
  SUBTYPE OF (managed_design_object);
  placement_fixed : BOOLEAN;
  transformation   : cartesian_transformation_operator_2d;
DERIVE
  substrate_location : BOOLEAN := substrate_in_assembly(SELF);
  the_context : cartesian_coordinate_system :=
referencing_position.assembly_model\shape_definition.geometric_context;
INVERSE
  referencing_position : component_assembly_2d_position FOR transformation;
WHERE
  WR1: NOT(substrate_location) OR (transformation.mirror = FALSE);
END_ENTITY; -- component_2d_location
( *

```

4.2.109 Component_2d_stacked_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_2d_stacked_location is a type of Component_2d_location (see 4.2.108).

The data associated with a Component_2d_stacked_location are the following:

— mounting_joint.

4.2.109.1 mounting_joint

The mounting_joint specifies an Assembly_joint (see 4.2.41) for the Component_2d_stacked_location.

NOTE EXPRESS specification:

```

*)
ENTITY component_2d_stacked_location
  SUBTYPE OF (component_2d_location);
  mounting_joint : assembly_joint;
--tt tc1 wd9
WHERE
(* tt tc1 wd13 deleted
  WR1: NOT EXISTS(mounting_surface);
*)
  WR1: SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
    'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE']) *
    TYPEOF(mounting_joint.assembly_features_1)) = 0;
  WR2: SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
    'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE']) *
    TYPEOF(mounting_joint.assembly_features_2)) = 0;
  WR3: SELF\component_2d_location.substrate_location = FALSE;
END_ENTITY; -- component_2d_stacked_location

(*

```

4.2.110 Component_2d_surface_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_2d_surface_location is a type of Component_2d_location (see 4.2.108). The Component_2d_surface_location places components on the external surface of a substrate.

NOTE Connection_zone_interface_plane_relationship (see 4.2.188) and Connection_zone_based_assembly_joint (see 4.2.186) may be used to provide more details if needed by a receiving organization.

The data associated with a Component_2d_surface_location are the following:

— mounting_surface_assembly_joint.

4.2.110.1 mounting_surface_assembly_joint

The mounting_surface_assembly_joint specifies the Assembly_joint (see 4.2.41) that joins the Component_mounting_feature (see 4.2.129) of the component being located to the associated mating component feature. The surface the mating component feature is associated with is part of the orientation description for the Component_2d_surface_location.

NOTE EXPRESS specification:

```

*)
ENTITY component_2d_surface_location
  SUBTYPE OF (component_2d_location);
    mounting_surface_assembly_joint : assembly_joint;
WHERE
  WR1: SELF\component_2d_location.substrate_location = FALSE;
  WR2: SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
    'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE']) *
    TYPEOF(mounting_surface_assembly_joint.assembly_features_1)) &#62;= 0;
END_ENTITY; -- component_2d_surface_location

```

(*

4.2.111 Component_3d_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_3d_location is a type of Managed_design_object (see 4.2.524). A Component_3d_location is the translation and orientation in three dimensions for the positioning of the Assembly_component (see 4.2.34). This Application Object requires Component_assembly_3d_position (see 4.2.113) for completeness.

The data associated with a Component_3d_location are the following:

- transformation;
- placement_fixed.

4.2.111.1 transformation

The transformation specifies the role of the Axis_placement_3d (see 4.2.58) for the Component_3d_location.

4.2.111.2 placement_fixed

The placement_fixed specifies whether or not the Component_3d_location is fixed. A value of TRUE indicates that the Component_3d_location is fixed. A value of FALSE indicates that the Component_3d_location is not fixed.

NOTE EXPRESS specification:

```

*)
ENTITY component_3d_location
  SUBTYPE OF (managed_design_object);
    transformation : axis_placement_3d;
    placement_fixed : BOOLEAN;
END_ENTITY; -- component_3d_location

( *
```

4.2.112 Component_assembly_2d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_assembly_2d_position is a type of Managed_design_object (see 4.2.524). A Component_assembly_2d_position is the placement and orientation for the positioning of the Assembly_component_2d_shape (see 4.2.35) in the Physical_unit_planar_shape (see 4.2.670) specified by the Next_higher_assembly_relationship (see 4.2.556).

The data associated with a Component_assembly_2d_position are the following:

- assembly_model;
- component;
- component_model;
- transformation.

4.2.112.1 assembly_model

The assembly_model specifies the role of the Physical_unit_planar_shape (see 4.2.670) for the Component_assembly_2d_position.

4.2.112.2 component

The component specifies the role of the Next_higher_assembly_relationship (see 4.2.556) for the Component_assembly_2d_position. The Next_higher_assembly_relationship specifies the actual component being placed.

4.2.112.3 component_model

The component_model specifies the role of the Assembly_component_2d_shape (see 4.2.35) for the Component_assembly_2d_position.

4.2.112.4 transformation

The transformation specifies the role of the Component_2d_location (see 4.2.108) for the Component_assembly_2d_position.

NOTE This application object is included to support exchange where a CAD geometric model is constrained to be two-dimensional models, and there is additional non-geometric characteristic information available.

NOTE EXPRESS specification:

```

*)
ENTITY component_assembly_2d_position
  SUBTYPE OF (managed_design_object);
    component_model : assembly_component_2d_shape;
    transformation   : component_2d_location;
    assembly_model   : physical_unit_planar_shape;
    component        : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_2d_position

( *
```

4.2.113 Component_assembly_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_assembly_3d_position is a type of Managed_design_object (see 4.2.524). A Component_assembly_3d_position is the placement location and orientation in three dimensions for the positioning of the Assembly_component_3d_shape (see 4.2.36) in the Physical_unit_3d_shape (see 4.2.666) specified by the Next_higher_assembly_relationship (see 4.2.556).

The data associated with a Component_assembly_3d_position are the following:

- assembly_model;
- component;
- component_model;
- transformation.

4.2.113.1 assembly_model

The assembly_model specifies the role of the Physical_unit_3d_shape (see 4.2.666) for the Component_assembly_3d_position.

4.2.113.2 component

The component specifies the role of the Next_higher_assembly_relationship (see 4.2.556) for the Component_assembly_3d_position. The Next_higher_assembly_relationship specifies the actual component being placed.

4.2.113.3 component_model

The component_model specifies the role of the Assembly_component_3d_shape (see 4.2.36) for the Component_assembly_3d_position.

4.2.113.4 transformation

The transformation specifies the role of the Component_3d_location (see 4.2.111) for the Component_assembly_3d_position.

NOTE EXPRESS specification:

```

*)
ENTITY component_assembly_3d_position
  SUBTYPE OF (managed_design_object);
    component_model : assembly_component_3d_shape;
    transformation  : component_3d_location;
    assembly_model  : physical_unit_3d_shape;
    component       : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_3d_position

( *
```

4.2.114 Component_assembly_non_planar_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_assembly_non_planar_position is a type of Managed_design_object (see 4.2.524). A Component_assembly_non_planar_position is the placement location and orientation in three dimensions for the positioning of the Assembly_component_non_planar_shape (see 4.2.37) in the Physical_unit_3d_shape (see 4.2.666) specified by the Next_higher_assembly_relationship (see 4.2.556).

The data associated with a `Component_assembly_non_planar_position` are the following:

- `assembly_model`;
- `component`;
- `component_model`;
- `transformation`.

4.2.114.1 assembly_model

The `assembly_model` specifies the role of the `Physical_unit_3d_shape` (see 4.2.666) for the `Component_assembly_non_planar_position`.

4.2.114.2 component

The `component` specifies the role of the `Next_higher_assembly_relationship` (see 4.2.556) for the `Component_assembly_non_planar_position`. The `Next_higher_assembly_relationship` specifies the actual component being placed.

4.2.114.3 component_model

The `component_model` specifies the role of the `Assembly_component_non_planar_shape` (see 4.2.37) for the `Component_assembly_non_planar_position`.

4.2.114.4 transformation

The `transformation` specifies the role of the `Component_non_planar_location` (see 4.2.130) for the `Component_assembly_non_planar_position`.

NOTE EXPRESS specification:

```

*)
ENTITY component_assembly_non_planar_position
  SUBTYPE OF (managed_design_object);
    component_model : assembly_component_non_planar_shape;
    transformation  : component_non_planar_location;
    assembly_model  : physical_unit_3d_shape;
    component       : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_non_planar_position
( *
```

4.2.115 Component_based_annotation_symbol_placement_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_based_annotation_symbol_placement_relationship` is a type of `Component_feature_to_layout_feature_relationship` (see 4.2.122). A `Component_based_annotation_symbol_placement_relationship` is the information that identifies a `Special_symbol_laminate_component` (see 4.2.834) and is to be placed in the context of the design. This information overrides any such information that may be provided in the library.

The data associated with a `Component_based_annotation_symbol_placement_relationship` are the following:

— `related_stratum_concept`.

NOTE This information is used to provide a means to reposition occurrences of polarization etch symbols, such as are used for polarized capacitors and diodes. The placement information is provided by the `Next_higher_assembly_relationship` for the occurrence of the etch symbol.

4.2.115.1 `related_stratum_concept`

The `related_stratum_concept` specifies the role of the `Special_symbol_laminate_component` (see 4.2.834) for the `Component_based_annotation_symbol_placement_relationship`.

NOTE EXPRESS specification:

```
*)
ENTITY component_based_annotation_symbol_placement_relationship
  SUBTYPE OF (component_feature_to_layout_feature_relationship);
  SELF\component_feature_to_layout_feature_relationship.related_stratum_concept :
  special_symbol_laminate_component;
END_ENTITY; -- component_based_annotation_symbol_placement_relationship

( *
```

4.2.116 `Component_based_annotation_text_relationship`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_based_annotation_text_relationship` is a type of `Component_feature_to_layout_feature_relationship` (see 4.2.122). A `Component_based_annotation_text_relationship` is the information that specifies an explicit text message based on the component, where the message content is either the value of the reference to the component's locator or the value of the terminal pin identifier.

NOTE This information permits product data stored in standard exchange files to serve as messages that are presented to the user in the medium of the Stratum (see 4.2.845), such as the text of the locator reference or the terminal pin name. This information is not intended to be populated for general etched characters on the board.

The data associated with a `Component_based_annotation_text_relationship` are the following:

- `message_type`;
- `related_stratum_concept`.

4.2.116.1 `message_type`

The `message_type` specifies either a component locator reference or a terminal locator for the `Component_based_annotation_text_relationship`. The `message_type` is a discriminator for determining which data to use from the attribute `relating_feature` that is inherited from the `Application Object Component_feature_to_layout_feature_relationship` (see 4.2.122). The useful options are either `Component_external_reference` (see 4.2.117) or `Component_feature_external_reference` (see 4.2.119). Either of these options will require a query based on the attribute `assembly_design_requirement` of the design which this `Component_feature_to_layout_feature_relationship` is helping to define. For the component locator reference option, two additional queries are required, one to determine the `Next_higher_assembly_relationship` (see 4.2.556) and the second to determine the `Assembly_component` (see 4.2.34). The component locator reference is the attribute `reference_designation` of the returned instance of `Next_higher_assembly_relationship`. The string content of that attribute shall be the string content of the message of the `related_stratum_concept`. For the terminal locator option, additional queries are necessary, continuing with the `Assembly_component` and terminating at the relevant `Part_terminal_external_reference` (see 4.2.631) with an intermediate query of `Component_feature` (see 4.2.118). The string content of the name attribute of `Part_terminal_external_reference` shall be the string content of the message of the `related_stratum_concept`.

NOTE 1 The attribute `message_type` is not a value of data. It is a declaration of the type of `Component_based_annotation_text_relationship`.

NOTE 2 The data contained in the `reference_designation` attribute in a member of `Next_higher_assembly_relationship` (see 4.2.556) is an instance of a message of one type.

NOTE 3 The data contained in the `name` attribute in a member of `Package_terminal` (see 4.2.581) is an instance of a message of the second type.

4.2.116.2 `related_stratum_concept`

The `related_stratum_concept` specifies the role of the `Laminate_text_string_component` (see 4.2.489) for the `Component_based_annotation_text_relationship`. There is no explicit geometric relationship defined in this part of ISO 10303 between the design element referenced by the message presented by the `related_stratum_concept` and the `related_stratum_concept` itself. Only the `related_stratum_concept` is positioned in the current design. The component or feature referenced by the message in the attribute `related_stratum_concept` is defined in a different design that is an `Assembly_module` (see 4.2.43).

NOTE This is the actual physical message, as realized by the fabrication process. The position of the corresponding elements in the two designs are usually defined by an end user using a layout tool which presents a view of the data from the assembly in the layout tool for design efficiency.

NOTE EXPRESS specification:

```

*)
ENTITY component_based_annotation_text_relationship
  SUBTYPE OF (component_feature_to_layout_feature_relationship);
  SELF\component_feature_to_layout_feature_relationship.related_stratum_concept :
  laminate_text_string_component;
  message_type :
  component_based_message_type;
END_ENTITY; -- component_based_annotation_text_relationship

( *
```

4.2.117 Component_external_reference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_external_reference is a type of Managed_design_object (see 4.2.524). A Component_external_reference is an unambiguous reference to a Next_higher_assembly_relationship (see 4.2.556) in a different design.

EXAMPLE 1 Items in an Interconnect_module (see 4.2.440) design view may need to reference component to assembly relationships in the assembly in order to unambiguously identify source of layout requirements.

The data associated with a Component_external_reference are the following:

- design_owner;
- part_number;
- revision_code;
- product_definition_id;
- reference_designation.

4.2.117.1 design_owner

The design_owner specifies one role of a string for the Component_external_reference. The design_owner is the organization that maintains the registration document for the part_number for the reference assembly. Usually it is the design authority.

4.2.117.2 part_number

The part_number specifies one role of a string for the Component_external_reference. The part_number identifies the reference assembly in the context of the design_owner.

4.2.117.3 revision_code

The revision_code specifies one role of a string for the Component_external_reference. The revision_code identifies the version of the reference assembly.

NOTE Usually the revision_code and part number identify the original assembly for which an interconnect is designed.

4.2.117.4 product_definition_id

The product_definition_id specifies one role of a string for the Component_external_reference. The product_definition_id identifies the original Ee_product_definition for the assembly. If not available in the originating system, the string "UNKNOWN" shall be provided.

4.2.117.5 reference_designation

The reference_designation specifies one role of a string for the Component_external_reference. The reference_designation uniquely identifies the externally referenced component in the reference assembly.

NOTE EXPRESS specification:

```

*)
ENTITY component_external_reference
  SUBTYPE OF (managed_design_object);
    design_owner      : STRING;
    part_number       : STRING;
    revision_code     : STRING;
    product_definition_id : STRING;
    reference_designation : STRING;
END_ENTITY; -- component_external_reference

( *
```

4.2.118 Component_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_feature is a type of Managed_design_object (see 4.2.524). A Component_feature may be one of a Component_interface_terminal (see 4.2.125), a Component_terminal (see 4.2.137), a Component_termination_passage_interface_terminal (see 4.2.148), a Land_interface_terminal (see 4.2.491), a Non_functional_land_interface_terminal (see 4.2.563), or a Printed_connector_component_interface_terminal (see 4.2.725), and in addition may be a Physical_component_feature (see 4.2.649), and in addition may be an Interconnect_module_component_surface_feature (see 4.2.444). The location of the Component_feature in the assembly model is derived from the placement of the associated_component in the assembly model and the placement of the definition model in the usage_definition model of that associated_component.

The data associated with a `Component_feature` are the following:

- `associated_component`;
- `defined_product_feature`;
- `definition`.

4.2.118.1 associated_component

The `associated_component` specifies the role of the `Assembly_component` (see 4.2.34) for the `Component_feature`.

NOTE The subtypes of `Component_feature` specify specializations of `Assembly_component` that are consistent with the subtype of `definition`.

4.2.118.2 defined_product_feature

The `defined_product_feature` is an inverse relationship that specifies that there shall be no more than one `Component_feature_to_physical_usage_view_assignment` (see 4.2.123) that specifies the `Component_feature` as its `assigned_design_object`. The `defined_product_feature` need not be specified for a particular `Component_feature`.

4.2.118.3 definition

The `definition` specifies either a `Component_termination_passage_template_terminal` (see 4.2.153), a `Land_template_terminal` (see 4.2.496), a `Part_feature` (see 4.2.609), a `Part_terminal_external_reference` (see 4.2.631), a `Printed_part_template_terminal` (see 4.2.737), or a `Via_template_terminal` (see 4.2.953) for this `Component_feature`. The `definition` need not be specified for a particular `Component_feature`.

Constraints:

UR1: The combination of the `definition` and the `associated_component` shall be unique within a population of `Component_features`.

WR1: The `Physical_unit` or `Part_template` specified by the `associated_component` as its `usage_definition` shall be the same `Physical_unit` or `Part_template` specified by the `definition` as its `associated_definition`.

WR2: For each `Component_feature` there shall be no more than one `Assembly_joint` that specifies it as an `assembly_feature`.

NOTE Parameters may be assigned to `Component_feature` by the `Planned_characteristic` object through the `Assembly_requirements_allocation`.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature
--tt tc1 wd12 modified
  SUPERTYPE OF (ONEOF (physical_component_feature,
    laminate_component_feature))
  SUBTYPE OF (managed_design_object);
  definition          : OPTIONAL part_feature_or_template_feature;
  associated_component : assembly_component;
  INVERSE
    defined_product_feature :
SET [0:1] OF component_feature_to_physical_usage_view_assignment FOR
assigned_design_object;
  UNIQUE
    UR1 : definition, associated_component;
END_ENTITY; -- component_feature

--tt tc1 wd12 added
(

```

4.2.119 Component_feature_external_reference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_feature_external_reference is a type of Managed_design_object (see 4.2.524). A Component_feature_external_reference is an unambiguous reference to a Component_feature (see 4.2.118) in a different design.

EXAMPLE 1 Items in an Interconnect_module (see 4.2.440) design view may need to reference component features in a reference assembly in order to unambiguously identify source of layout requirements.

The data associated with a Component_feature_external_reference are the following:

- associated_component;
- name.

4.2.119.1 associated_component

The associated_component specifies the role for a Component_external_reference (see 4.2.117) for the Component_feature_external_reference.

4.2.119.2 name

The name specifies the role of a string for the Component_feature_external_reference.

Constraints:

UR1: Each member of Component_feature_external_reference shall have a unique combination of attributes.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature_external_reference
  SUBTYPE OF (managed_design_object);
    associated_component : component_external_reference;
    name                 : STRING;
UNIQUE
  UR1 : associated_component, name;
END_ENTITY; -- component_feature_external_reference

( *
```

4.2.120 Component_feature_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_feature_make_from_relationship is a type of Managed_design_object (see 4.2.524). A Component_feature_make_from_relationship is an association between two Component_feature (see 4.2.118), where one Component_feature is used as a basis for the development of another Component_feature.

The data associated with a Component_feature_make_from_relationship are the following:

- resultant_feature;
- reusable_feature.

4.2.120.1 resultant_feature

The resultant_feature specifies the Component_feature (see 4.2.118) that results from modifying or reusing the reusable_feature in the Component_feature_make_from_relationship.

4.2.120.2 reusable_feature

The reusable_feature specifies the Component_feature (see 4.2.118) that is modified or reused to become the resultant_feature in the Component_feature_make_from_relationship.

Constraints:

WR1: The resultant_feature shall not be the reusable_feature either directly or indirectly.

WR2: The type of the resultant_feature shall be the same as the type of the reusable_feature.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_feature : component_feature;
    resultant_feature : component_feature;
END_ENTITY; -- component_feature_make_from_relationship

( *
```

4.2.121 Component_feature_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_feature_relationship is a type of Managed_design_object (see 4.2.524). A Component_feature_relationship is the association between two Component_feature (see 4.2.118).

The data associated with a Component_feature_relationship are the following:

- related_component_feature;
- relating_component_feature.

4.2.121.1 related_component_feature

The related_component_feature specifies the dependent Component_feature (see 4.2.118) for the Component_feature_relationship.

4.2.121.2 relating_component_feature

The relating_component_feature specifies the independent Component_feature (see 4.2.118) for the Component_feature_relationship.

Constraints:

WR1: The related_component_feature shall not be the relating_component_feature either directly or indirectly.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature_relationship
  SUBTYPE OF (managed_design_object);
    related_component_feature : component_feature;
    relating_component_feature : component_feature;
END_ENTITY; -- component_feature_relationship

( *
```

4.2.122 Component_feature_to_layout_feature_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_feature_to_layout_feature_relationship is a type of Managed_design_object (see 4.2.524). A Component_feature_to_layout_feature_relationship may be a Component_based_annotation_symbol_placement_relationship (see 4.2.115) or a Component_based_annotation_text_relationship (see 4.2.116). A Component_feature_to_layout_feature_relationship is the association between a component or feature of a component and a layout feature, based upon a requirement. All features related with this relationship shall be in the same design unless the reference is to a Component_external_reference (see 4.2.117) or to a Component_feature_external_reference (see 4.2.119).

The data associated with a Component_feature_to_layout_feature_relationship are the following:

- related_stratum_concept;
- relating_feature;
- requirement.

4.2.122.1 related_stratum_concept

The related_stratum_concept specifies either an Inter_stratum_feature (see 4.2.432), a Laminate_component (see 4.2.483), a Stratum (see 4.2.845), a Stratum_feature (see 4.2.851), or a Stratum_surface (see 4.2.868) to which the relating_feature is related for the Component_feature_to_layout_feature_relationship.

4.2.122.2 relating_feature

The relating_feature specifies an Assembly_component (see 4.2.34) or a Component_feature (see 4.2.118) or a Component_feature_external_reference or a Component_external_reference for the Component_feature_to_layout_feature_relationship.

4.2.122.3 requirement

The requirement specifies the `Ee_requirement_occurrence` (see 4.2.330) that causes this `Component_feature_to_layout_feature_relationship` to be needed. The requirement need not be specified for a particular `Component_feature_to_layout_feature_relationship`.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature_to_layout_feature_relationship
  SUPERTYPE OF (ONEOF (component_based_annotation_text_relationship,
    component_based_annotation_symbol_placement_relationship))
  SUBTYPE OF (managed_design_object);
  related_stratum_concept : stratum_concept;
  relating_feature        : component_or_component_feature;
  requirement             : OPTIONAL ee_requirement_occurrence;
END_ENTITY; -- component_feature_to_layout_feature_relationship

--tt tc1 wd9 modified
( *
```

4.2.123 Component_feature_to_physical_usage_view_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_feature_to_physical_usage_view_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Component_feature_to_physical_usage_view_assignment` is the association between the `Part_feature` (see 4.2.609) and the `Component_feature` (see 4.2.118). The purpose for this assignment is to link the product data in the design view to the product data in the usage view for traceability purposes.

NOTE 1 In an archive, both a design view and a usage view are archived together, with this relationship populated as necessary to provide linkages between the features in the usage view and the features in the design view. Without this linkage populated, there is no way to guarantee traceability between a design view and a usage view, without evaluating the design view and the usage view in detail.

NOTE 2 In the case where this part of ISO 10303 is used to exchange or store designs that may be used in higher level assemblies, the usage view is provided to the designers of those higher level assemblies, and this entity allows the interface definitions to the next level assemblies to be carefully controlled. Note that the usage view is an as-designed view. The requirements unit of functionality in this part of ISO 10303 supports the as-required view of the interface to the next higher assembly in a top-down design paradigm. This part of ISO 10303 supports explicitly exchanging those data separately so that traceability is provided from requirements to as-designed life-cycle states in the same exchange file.

NOTE 3 This information is provided for the case where a `Component_feature` is suitable directly as a feature of the assembly.

NOTE 4 Most `Component_feature` types will be interface types.

The data associated with a `Component_feature_to_physical_usage_view_assignment` are the following:

- `assigned_design_object`;
- `defined_part_feature`.

4.2.123.1 `assigned_design_object`

The `assigned_design_object` specifies the role of the `Component_feature` (see 4.2.118) for the `Component_feature_to_physical_usage_view_assignment`.

4.2.123.2 `defined_part_feature`

The `defined_part_feature` specifies the role of the `Part_feature` (see 4.2.609) for the `Component_feature_to_physical_usage_view_assignment`.

NOTE EXPRESS specification:

```

*)
ENTITY component_feature_to_physical_usage_view_assignment
  SUPERTYPE OF (ONEOF (
    component_terminal_to_assembly_module_join_terminal_assignment,
    component_terminal_to_assembly_module_interface_terminal_assignment,
    component_terminal_to_interconnect_module_join_terminal_assignment,
    component_terminal_to_interconnect_module_interface_terminal_assignment,
    component_terminal_to_pca_terminal_assignment))
  SUBTYPE OF (managed_design_object);
    defined_part_feature : part_feature;
    assigned_design_object : component_feature;
END_ENTITY; -- component_feature_to_physical_usage_view_assignment

( *
```

4.2.124 `Component_group_assignment`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_group_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Component_group_assignment` is the relationship between an `Assembly_component` (see 4.2.34) and the `Placement_group` (see 4.2.674) to which it is assigned. A `Component_group_assignment` is an incomplete concept by itself and is intended for use with a `Placement_group_area_assignment` (see 4.2.675) or a `Component_placement_restriction_assignment` (see 4.2.135).

The data associated with a `Component_group_assignment` are the following:

- `assigned_component`;
- `associated_group`.

4.2.124.1 `assigned_component`

The `assigned_component` specifies the `Assembly_component` (see 4.2.34) in the `Component_group_assignment`.

4.2.124.2 `associated_group`

The `associated_group` specifies the `Placement_group` (see 4.2.674) in the `Component_group_assignment`.

NOTE EXPRESS specification:

```

*)
ENTITY component_group_assignment
  SUBTYPE OF (managed_design_object);
    associated_group    : placement_group;
    assigned_component : assembly_component;
END_ENTITY; -- component_group_assignment

( *
```

4.2.125 `Component_interface_terminal`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_interface_terminal` is a type of `Component_feature` (see 4.2.118). A `Component_interface_terminal` is one of an `Interconnect_component_interface_terminal` (see 4.2.438) or a `Packaged_connector_component_interface_terminal` (see 4.2.589). A `Component_interface_terminal` is one termination of an interface component.

EXAMPLE 1 An example of this is one terminal on a socket or connector, available for mating to a terminal of an external connector or to a terminal of a component installed in the socket.

The data associated with a `Component_interface_terminal` are the following:

- `associated_component`.

4.2.125.1 `associated_component`

The `associated_component` specifies the role of the `Interface_component` (see 4.2.468) for the `Component_interface_terminal`.

NOTE EXPRESS specification:

```

*)
ENTITY component_interface_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (
    packaged_connector_component_interface_terminal,
    interconnect_component_interface_terminal))
  SUBTYPE OF (physical_component_feature);
--tt tc1 wd12 modified
--tt tc1 wd9 modified
  SELF\component_feature.associated_component : interface_component;
END_ENTITY; -- component_interface_terminal

```

(*

4.2.126 Component_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_make_from_relationship is a type of Managed_design_object (see 4.2.524). A Component_make_from_relationship is the association between one Assembly_composition_relationship (see 4.2.39) and another Assembly_composition_relationship, where one Assembly_composition_relationship results from full or partial reuse of the other. A Component_make_from_relationship may be a Laminate_group_component_make_from_relationship.

NOTE The main use of this concept is design reuse where the modification consists of the fact that the Assembly_composition_relationship is in a different assembly from the original design, or where the original design is a reference design.

The data associated with a Component_make_from_relationship are the following:

- resultant_component;
- reusable_component.

4.2.126.1 resultant_component

The resultant_component specifies the Assembly_composition_relationship (see 4.2.39) for the Assembly_component (see 4.2.34) that results from the reuse of the reusable_component in the Component_make_from_relationship.

4.2.126.2 reusable_component

The reusable_component specifies the Assembly_composition_relationship (see 4.2.39) for the Assembly_component (see 4.2.34) that is reused to create the resultant_component in the Component_make_from_relationship.

Constraints:

WR1: The reusable_component shall not be an ancestor, either directly or indirectly of the resultant_component. The inverse shall also be true.

WR2: The reusable_component shall not be the resultant_component.

NOTE EXPRESS specification:

```

*)
ENTITY component_make_from_relationship
  SUPERTYPE OF(laminate_group_component_make_from_relationship)
  SUBTYPE OF (managed_design_object);
  reusable_component : assembly_component;
  resultant_component : assembly_component;
WHERE
  WR1: resultant_component &#60;&#62; reusable_component;
END_ENTITY; -- component_make_from_relationship

( *
```

4.2.127 Component_mating_constraint_condition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_mating_constraint_condition is a type of Managed_design_object (see 4.2.524). A Component_mating_constraint_condition identifies the degrees of freedom that have been constrained by a set of members of Assembly_joint (see 4.2.41). A Component_mating_constraint_condition provides "x","y","z" as the set of axes, and "u","v","w" as the set of rotations about those axes. If an axis is included in the set, then translation along that axis is constrained. If a rotation is included in the set, then rotation about the corresponding axis is constrained. The pairs of axis,rotation are: x,u; y,v;z,w. The axes are referenced in the local coordinate system of the component referenced by the related attribute of the members of Assembly_joint referenced by constraining_joints. Exactly one member of Assembly_component shall be referenced by the related attribute of members of Assembly_joint referenced by constraining_joints.

The data associated with a Component_mating_constraint_condition are the following:

- constraining_joint;
- constrained_degrees_of_freedom;
- context;
- name.

4.2.127.1 constraining_joint

The `constraining_joint` specifies the `Assembly_joint` for the `Component_mating_constraint_condition`. There shall be at least one member of `constraining_joint` and there shall be no more than three members of `constraining_joint`.

4.2.127.2 constrained_degrees_of_freedom

The `constrained_degrees_of_freedom` specifies the identifier of the degrees of freedom that are constrained. There shall be at least one and there shall be no more than five degrees of freedom specified.

4.2.127.3 context

The `of_context` specifies the role of the `Physical_unit_design_view` (see 4.2.667) for the `Component_mating_constraint_condition`.

4.2.127.4 name

The `name` specifies the identifier of the `Component_mating_constraint_condition` in the context of the `Physical_unit_design_view`.

NOTE EXPRESS specification:

```

*)
ENTITY component_mating_constraint_condition
  SUBTYPE OF (managed_design_object);
  constraining_joint : SET[1:3] OF assembly_joint;
  constrained_degree_of_freedom : SET[1:5] OF degree_of_freedom;
  of_context : physical_unit_design_view;
  name : ee_name;
UNIQUE
  UR1: name, of_context;
END_ENTITY; -- component_mating_constraint_condition
( *
```

4.2.128 Component_mounting_clearance_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_mounting_clearance_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Component_mounting_clearance_relationship` is the relationship between the features of one component and another in their next assembly, with respect to the clearance between them.

NOTE 1 This clearance is required when a specific component requires a minimum distance from the printed circuit board or another specific component.

NOTE 2 This clearance is often derived from the mechanism requirements of the factory equipment that assembles the components to their next assembly. The mechanical handling fingers that grasp and place a component require clearance from previously placed components in order to place the second component and release it from the mechanism.

The data associated with a `Component_mounting_clearance_relationship` are the following:

- `feature_relationship`;
- `maximum_mounting_clearance`;
- `minimum_mounting_clearance`.

4.2.128.1 `feature_relationship`

The `current_component` specifies the `Component_feature_relationship` (see 4.2.121) of the features of the components to be placed for the `Component_mounting_clearance_relationship`.

4.2.128.2 `maximum_mounting_clearance`

The `maximum_mounting_clearance` specifies the `Length_data_element` (see 4.2.511), of the maximum distance between the `current_component` and the `previously_placed_component`, for the `Component_mounting_clearance_relationship`. The `maximum_mounting_clearance` need not be specified for a particular `Component_mounting_clearance_relationship`. The value shall be interpreted as the specification to be satisfied by the assembly process.

4.2.128.3 `minimum_mounting_clearance`

The `minimum_mounting_clearance` specifies the `Length_data_element` (see 4.2.511), of the minimum distance between the `current_component` and the `previously_placed_component`, for the `Component_mounting_clearance_relationship`. The value shall be interpreted as the specification to be satisfied by the assembly process.

Constraints:

WR1: The `feature_relationship` shall relate two features in the same design but not of the same component.

WR2: A `Datum_plane` shall be defined for each of the `current_component` and of the `previously_placed_component`.

WR3: The length shall be measured between the two `Datum_planes`.

WR4: Both components shall be in the same version of the product.

WR5: Each Datum_plane shall be based on physical features referenced by the feature_relationship.

NOTE EXPRESS specification:

```

*)
--tt tc1 wd13 modified
ENTITY component_mounting_clearance_relationship
  SUBTYPE OF (managed_design_object);
    feature_relationship      : component_feature_relationship;
    minimum_mounting_clearance : length_data_element;
    maximum_mounting_clearance : OPTIONAL length_data_element;
END_ENTITY; -- component_mounting_clearance_relationship

--tt tc1 wd8 added
( *
```

4.2.129 Component_mounting_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_mounting_feature is a type of Physical_component_feature (see 4.2.649).

The data associated with a Component_mounting_feature are the following:

— definition.

4.2.129.1 definition

The definition specifies the role of the Part_mounting_feature (see 4.2.618) for the Component_mounting_feature.

NOTE EXPRESS specification:

```

*)
ENTITY component_mounting_feature
  SUBTYPE OF (physical_component_feature);
    SELF\component_feature.definition : part_mounting_feature;
END_ENTITY; -- component_mounting_feature

( *
```

4.2.130 Component_non_planar_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_non_planar_location` is a type of `Managed_design_object` (see 4.2.524). A `Component_non_planar_location` is the translation and orientation in three dimensions for the positioning of the `Assembly_component` (see 4.2.34). This Application Object requires `Component_assembly_3d_position` (see 4.2.113) for completeness.

The data associated with a `Component_non_planar_location` are the following:

- `transformation`;
- `placement_fixed`.

4.2.130.1 transformation

The `transformation` specifies the role of the `Axis_placement_3d` (see 4.2.58) for the `Component_non_planar_location`.

4.2.130.2 placement_fixed

The `placement_fixed` specifies whether or not the `Component_non_planar_location` is fixed. A value of `TRUE` indicates that the `Component_non_planar_location` is fixed. A value of `FALSE` indicates that the `Component_non_planar_location` is not fixed.

NOTE EXPRESS specification:

```
*)
ENTITY component_non_planar_location
  SUBTYPE OF (managed_design_object);
    transformation : axis_placement_3d;
    placement_fixed : BOOLEAN;
END_ENTITY; -- component_non_planar_location
(*
```

4.2.131 Component_overlap_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_overlap_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Component_overlap_relationship` is the relationship between one component and another in their next assembly, with respect to the physical overlap between them. This overlap is often derived from the mechanism requirements of the factory equipment that assembles the components to their next assembly. The sequence of assembly is established by the `previously_placed_component` and the `current_component`, in turn, and the minimum clearance between the two components is taken into consideration during assembly.

The data associated with a `Component_overlap_relationship` are the following:

- `current_component`;
- `maximum_added_clearance`;
- `minimum_added_clearance`;
- `previously_placed_component`.

4.2.131.1 `current_component`

The `current_component` specifies the `Next_higher_assembly_relationship` (see 4.2.556), of the component to be placed next, for the `Component_overlap_relationship`.

4.2.131.2 `maximum_added_clearance`

The `maximum_added_clearance` specifies the `Length_data_element` (see 4.2.511) of the maximum distance between the `current_component` and the `previously_placed_component` for the `Component_overlap_relationship`. The `maximum_added_clearance` need not be specified for a particular `Component_overlap_relationship`.

4.2.131.3 `minimum_added_clearance`

The `minimum_added_clearance` specifies the `Length_data_element` (see 4.2.511) of the minimum distance between the `current_component` and the `previously_placed_component` for the `Component_overlap_relationship`.

4.2.131.4 `previously_placed_component`

The `previously_placed_component` specifies the `Next_higher_assembly_relationship` (see 4.2.556), of the component previously placed for the `Component_overlap_relationship`.

Constraints:

WR1: The `previously_placed_component` shall not be the `current_component`.

WR2: Both components shall be in the same version of the assembly.

WR3: A `Datum_plane` shall be defined for each of the `current_component` and `previously_placed_component`.

WR4: The length shall be measured between the two `Datum_planes`.

WR5: Each `Datum_plane` shall be based on a physical feature of the component.

NOTE EXPRESS specification:

```

*)
ENTITY component_overlap_relationship
  SUBTYPE OF (managed_design_object);
    previously_placed_component : next_higher_assembly_relationship;
    current_component           : next_higher_assembly_relationship;
    minimum_added_clearance     : length_data_element;
    maximum_added_clearance     : OPTIONAL length_data_element;
END_ENTITY; -- component_overlap_relationship

( *

```

4.2.132 Component_part_2d_non_planar_geometric_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_part_2d_non_planar_geometric_representation_relationship is a type of Geometric_representation_relationship (see 4.2.418). A Component_part_2d_non_planar_geometric_representation_relationship is the association between a Part_template_planar_shape (see 4.2.628) or a Physical_unit_planar_shape (see 4.2.670) and the Assembly_component_2d_shape (see 4.2.35) that is the occurrence of that planar shape. If the type of the shape representation of the assembly is different from the type of the shape representation of the part, the Component_part_2d_non_planar_geometric_representation_relationship allows a processor to create a shape rep of the assembly type and establish an equivalency relationship between that shape representation and the part shape representation. If the shape representation of the assembly is of the same type as the part shape representation, the Component_part_2d_non_planar_geometric_representation_relationship shall reference the same shape representation as both the part_shape and the component_shape. That instance shall be a complex instance of the part shape and of the component shape.

The data associated with a Component_part_2d_non_planar_geometric_representation_relationship are the following:

- component_shape;
- part_shape.

4.2.132.1 component_shape

The component_shape specifies the role of the Assembly_component_non_planar_shape (see 4.2.37) for the Component_part_2d_non_planar_geometric_representation_relationship.

4.2.132.2 part_shape

The part_shape specifies the Part_template_planar_shape (see 4.2.628) for the Component_part_2d_non_planar_geometric_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY component_part_2d_non_planar_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_planar_shape :=
                      SELF\geometric_representation_relationship.shape_1;
    component_shape : assembly_component_non_planar_shape :=
                      SELF\geometric_representation_relationship.shape_2;
END_ENTITY; -- component_part_2d_non_planar_geometric_representation_relationship
( *

```

4.2.133 Component_part_2d_geometric_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_part_2d_geometric_representation_relationship is a type of Geometric_representation_relationship (see 4.2.418). A Component_part_2d_geometric_representation_relationship is the association between a Part_template_planar_shape (see 4.2.628) or a Physical_unit_planar_shape (see 4.2.670) and the Assembly_component_2d_shape (see 4.2.35) that is the occurrence of that planar shape. If the type of the shape representation of the assembly is different from the type of the shape representation of the part, the Component_part_2d_geometric_representation_relationship allows a processor to create a shape rep of the assembly type and establish an equivalency relationship between that shape representation and the part shape representation. If the shape representation of the assembly is of the same type as the part shape representation, the Component_part_2d_geometric_representation_relationship shall reference the same shape representation as both the part_shape and the component_shape. That instance shall be a complex instance of the part shape and of the component shape.

The data associated with a Component_part_2d_geometric_representation_relationship are the following:

- component_shape;
- part_shape.

4.2.133.1 component_shape

The component_shape specifies the role of the Assembly_component_2d_shape (see 4.2.35) for the Component_part_2d_geometric_representation_relationship.

4.2.133.2 part_shape

The part_shape specifies either the Part_template_planar_shape (see 4.2.628) or the Physical_unit_planar_shape (see 4.2.670) for the Component_part_2d_geometric_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY component_part_2d_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_or_physical_unit_2d_shape_select :=
                      SELF\geometric_representation_relationship.shape_1;
    component_shape : assembly_component_2d_shape :=
                      SELF\geometric_representation_relationship.shape_2;
END_ENTITY; -- component_part_2d_geometric_representation_relationship
(*

```

4.2.134 Component_part_3d_geometric_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_part_3d_geometric_representation_relationship is a type of Geometric_representation_relationship (see 4.2.418). A Component_part_3d_geometric_representation_relationship is the association between a Part_template_3d_shape (see 4.2.621) or a Physical_unit_3d_shape (see 4.2.666) and the Assembly_component_3d_shape (see 4.2.36) that is the occurrence of the three-dimensional shape. If the type of the shape representation of the assembly is different from the type of the shape representation of the part, the Component_part_3d_geometric_representation_relationship allows a processor to create a shape rep of the assembly type and establish an equivalency relationship between that shape representation and the part shape representation. If the shape representation of the assembly is of the same type as the part shape representation, the Component_part_3d_geometric_representation_relationship shall reference the same shape representation as both the part_shape and the component_shape. That instance shall be a complex instance consisting of a combination of a part shape and of a component shape.

The data associated with a Component_part_3d_geometric_representation_relationship are the following:

- component_shape;
- part_shape.

4.2.134.1 component_shape

The component_shape specifies the role of the Assembly_component_3d_shape (see 4.2.36) for the Component_part_3d_geometric_representation_relationship.

4.2.134.2 part_shape

The part_shape specifies the Part_template_3d_shape (see 4.2.621), or the Part_template_non_planar_2d_shape (see 4.2.625), or the Physical_unit_3d_shape (see 4.2.666) for the Component_part_3d_geometric_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY component_part_3d_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_or_physical_unit_3d_shape_select :=
                        SELF\geometric_representation_relationship.shape_1;
    component_shape : assembly_component_3d_shape :=
                        SELF\geometric_representation_relationship.shape_2;
END_ENTITY; -- component_part_3d_geometric_representation_relationship
( *
```

4.2.135 Component_placement_restriction_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_placement_restriction_assignment is a type of Managed_design_object (see 4.2.524). A Component_placement_restriction_assignment is the relationship between the area of an entire Interconnect_module_component (see 4.2.442) or a portion thereof and restrictions to placement of other components in that area due to an item in the design. The context of a Component_placement_restriction_assignment is the design of the assembly within which both the Interconnect_module_component and associated components are identified.

The data associated with a Component_placement_restriction_assignment are the following:

- area;
- components_permitted;
- maximum_mounting_clearance;
- maximum_negative_component_height;
- maximum_positive_component_height;
- minimum_mounting_clearance;
- requirement;
- volume.

4.2.135.1 area

The area specifies the role of the Mounting_restriction_area (see 4.2.550) for the Component_placement_restriction_assignment. The area is an area on an Interconnect_module_component.

4.2.135.2 components_permitted

The `components_permitted` specifies whether or not there is a restriction upon placement of components in an area for the `Component_placement_restriction_assignment`. The value of `TRUE` states that components are permitted; the value of `FALSE` states that components are not permitted to be placed in the defined area.

4.2.135.3 maximum_mounting_clearance

The `maximum_mounting_clearance` specifies the largest `Datum_based_length_measure` (see 4.2.219) for which the `Component_placement_restriction_assignment` is made. The `maximum_mounting_clearance` need not be specified for a particular `Component_placement_restriction_assignment`.

EXAMPLE 1 An example of the `maximum_mounting_clearance` attribute is for a printed circuit assembly with through-hole components, where some packages are required to be mounted other than flush with the surface of the printed circuit board, for purposes of cleaning or of thermal dissipation. The `maximum_mounting_clearance` is less frequently seen than the `minimum_mounting_clearance`, but it is included for completeness. When a maximum dimension is required, it is often described by the `maximum_positive_component_height` attribute, which takes into consideration the maximum material condition of the package as well as the clearance from the printed circuit board.

4.2.135.4 maximum_negative_component_height

The `maximum_negative_component_height` specifies the largest `Datum_based_length_measure` (see 4.2.219) in the negative direction from that Datum (see 4.2.214), for which the `Component_placement_restriction_assignment` is made. The `maximum_negative_component_height` need not be specified for a particular `Component_placement_restriction_assignment`.

EXAMPLE 1 An example of the `maximum_negative_component_height` attribute is a printed circuit assembly with through-hole components, where the maximum lead protrusion dimension is given. That dimension is the `maximum_negative_component_height`. Another example of the `maximum_negative_component_height` attribute is a printed circuit assembly with surface mounted components on both sides. The `maximum_negative_component_height` attribute is the distance between the established datum and the maximum material condition of the vertical dimension of the highest surface mount component on the secondary side of the printed circuit assembly.

4.2.135.5 maximum_positive_component_height

The `maximum_positive_component_height` specifies the largest `Datum_based_length_measure` (see 4.2.219) in the positive direction from the Datum (see 4.2.214), for which the `Component_placement_restriction_assignment` is made. The `maximum_positive_component_height` need not be specified for a particular `Component_placement_restriction_assignment`.

EXAMPLE 1 An example of the `maximum_positive_component_height` attribute is a printed circuit assembly where the maximum component height in the positive direction relative to a datum is given. This dimension pertains to the fit of the printed circuit assembly in its next assembly. That dimension is the `maximum_positive_component_height`.

4.2.135.6 minimum_mounting_clearance

The minimum_mounting_clearance specifies the smallest Datum_based_length_measure (see 4.2.219) for which the Component_placement_restriction_assignment is made. The minimum_mounting_clearance need not be specified for a particular Component_placement_restriction_assignment.

EXAMPLE 1 An example of the minimum_mounting_clearance attribute is for a printed circuit assembly with through-hole components, where some packages are required to be mounted other than flush with the surface of the printed circuit board for purposes of cleaning or of thermal dissipation. The minimum_mounting_clearance is more frequently seen than the maximum_mounting_clearance. The usual default clearance dimension is zero, meaning that the component package is intended to be mounted flush with the printed circuit board. Packages that often require a minimum_mounting_clearance include radial capacitors and certain other devices where the leads protrude from the bottom of the package, and where the package is not usually intended to be bonded directly to the printed circuit board.

4.2.135.7 requirement

The requirement specifies the role of the Item_restricted_requirement_occurrence (see 4.2.478) for the Component_placement_restriction_assignment.

4.2.135.8 volume

The volume specifies the role of the Mounting_restriction_volume (see 4.2.551) for the Component_placement_restriction_assignment. The volume is in the context of the assembly.

Constraints:

WR1: The direction of negative measurement is antiparallel to the normal of the surface of the Interconnect_module_component specified by area.mounting_surface.

WR2: The direction of positive measurement is parallel to the normal of the surface of the Interconnect_module_component specified by area.mounting_surface.

WR3: The location of the Datum for the maximum_mounting_clearance_relationship shall be on the same side of the Interconnect_module_component as that specified by area.mounting_surface.

WR4: The location of the Datum for the minimum_mounting_clearance_relationship shall be on the same side of the Interconnect_module_component as that specified by area.mounting_surface.

WR5: At least one of area or volume shall be provided.

NOTE EXPRESS specification:

```

*)
ENTITY component_placement_restriction_assignment
  SUBTYPE OF (managed_design_object);
    maximum_negative_component_height : OPTIONAL datum_based_length_measure;
    maximum_positive_component_height : OPTIONAL datum_based_length_measure;
    area                               : OPTIONAL mounting_restriction_area;
    volume                             : OPTIONAL mounting_restriction_volume;
    requirement                         : item_restricted_requirement_occurrence;
    components_permitted                : BOOLEAN;
    maximum_mounting_clearance          : OPTIONAL datum_based_length_measure;
    minimum_mounting_clearance          : OPTIONAL datum_based_length_measure;
WHERE
  WR1: EXISTS(area) OR EXISTS(volume);
END_ENTITY; -- component_placement_restriction_assignment

( *

```

4.2.136 Component_swap_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_swap_specification is a type of Change_design_object_management_relationship (see 4.2.95). The Component_swap_specification specifies which component has been swapped for which other component in a design; both the swapped-out and the swapped-in components are specified.

NOTE The Component_swap_specification supports the component identification portion of the gate swap specification. It is necessary when components, such as the microcircuits known as U1, U2, etc. are being swapped.

The data associated with a Component_swap_specification are the following:

- current_design_object;
- previous_design_object.

4.2.136.1 current_design_object

The current_design_object specifies the Design_functional_unit_allocation_to_assembly_component (see 4.2.263) for the resulting design object after swapping, for the Component_swap_specification.

4.2.136.2 previous_design_object

The previous_design_object specifies the Design_functional_unit_allocation_to_assembly_component (see 4.2.263) for the original design object before swapping, for the Component_swap_specification.

NOTE EXPRESS specification:

```

*)
ENTITY component_swap_specification
  SUBTYPE OF (change_design_object_management_relationship);
  SELF\change_design_object_management_relationship.current_design_object :
design_functional_unit_allocation_to_assembly_component;
  SELF\change_design_object_management_relationship.previous_design_object :
design_functional_unit_allocation_to_assembly_component;
END_ENTITY; -- component_swap_specification

( *

```

4.2.137 Component_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_terminal is a type of Component_feature (see 4.2.118). A Component_terminal is one of an Assembly_module_component_terminal (see 4.2.45), a Bare_die_component_terminal (see 4.2.62), a Component_termination_passage_join_terminal (see 4.2.149), a Conductive_interconnect_element_terminal (see 4.2.172), an Interconnect_component_join_terminal (see 4.2.439), an Interconnect_module_component_terminal (see 4.2.445), a Land_join_terminal (see 4.2.492), a Non_functional_land_join_terminal (see 4.2.564), a Packaged_component_join_terminal (see 4.2.586), a Printed_component_join_terminal (see 4.2.722), or a Via_terminal (see 4.2.954). A Component_terminal is the defined conductive shape for conductors from other components to access electrical functionality of the Assembly_component (see 4.2.34).

The data associated with a Component_terminal are the following:

- connected_node;
- global_swappable;
- local_swappable;
- node_implementation;
- swap-code.

4.2.137.1 connected_node

The connected_node is an inverse relationship that specifies the Physical_connectivity_definition (see 4.2.652) that specifies the Component_terminal as its associated_terminals. There shall be no more than one Physical_connectivity_definition for a Component_terminal. The connected_node need not be specified for a particular Component_terminal.

4.2.137.2 global_swappable

The global_swappable specifies a boolean value that indicates if the terminal can be exchanged within associated functional units of the same type.

4.2.137.3 local_swappable

The local_swappable specifies a boolean value that indicates if the terminal can be exchanged within the associated functional unit.

4.2.137.4 node_implementation

The node_implementation is an inverse relationship that specifies the Physical_network (see 4.2.660) that specifies the Component_terminal as its connectivity_requirement_element. There shall be no more than one Physical_network for a Component_terminal. The node_implementation need not be specified for a particular Component_terminal.

4.2.137.5 swap-code

The swap_code specifies the string that is the type of exchange that is valid.

Constraints:

IP1: Each Component_terminal that is referenced by a Physical_connectivity_definition as an associated_terminal shall not also be in the design view of the implementation of that Physical_connectivity_definition.

IP2: The node_implementation and the connected_node shall not both exist.

NOTE EXPRESS specification:

```
*)
ENTITY component_terminal
--tt tc1 wd12 modified
  ABSTRACT SUPERTYPE OF (ONEOF (
    physical_component_terminal,
    laminate_component_join_terminal))
  SUBTYPE OF (component_feature);
  local_swappable : OPTIONAL BOOLEAN;
  global_swappable : OPTIONAL BOOLEAN;
  swap_code : OPTIONAL STRING;
(* tt tc1 wd12 deleted INVERSE, moved to subtypes *)
END_ENTITY; -- component_terminal

--tt tc1 wd9 modified
(*
```

4.2.138 Component_terminal_to_assembly_module_interface_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_terminal_to_assembly_module_interface_terminal_assignment is a type of Component_feature_to_physical_usage_view_assignment (see 4.2.123).

NOTE EXPRESS specification:

```

*)
ENTITY component_terminal_to_assembly_module_interface_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
DERIVE
    usage_view_terminal : assembly_module_interface_terminal :=
  SELF\component_feature_to_physical_usage_view_assignment.
  defined_part_feature;
    assigned_terminal : pca_interface_select :=
  SELF\component_feature_to_physical_usage_view_assignment.
  assigned_design_object;

  UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_assembly_module_interface_terminal_assignment

--tt tc1 wd9 added
( *
```

4.2.139 Component_terminal_to_assembly_module_join_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_terminal_to_assembly_module_join_terminal_assignment is a type of Component_feature_to_physical_usage_view_assignment (see 4.2.123).

NOTE EXPRESS specification:

```

*)
ENTITY component_terminal_to_assembly_module_join_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
DERIVE
    usage_view_terminal : assembly_module_join_terminal :=
    SELF\component_feature_to_physical_usage_view_assignment.
    defined_part_feature;
    assigned_terminal :
    interconnect_module_macro_component_join_terminal :=
    SELF\component_feature_to_physical_usage_view_assignment.
    assigned_design_object;

    UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_assembly_module_join_terminal_assignment

--tt tc1 wd9 modified
(*)

```

4.2.140 Component_terminal_to_interconnect_module_interface_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_terminal_to_interconnect_module_interface_terminal_assignment is a type of Component_feature_to_physical_usage_view_assignment (see 4.2.123).

NOTE EXPRESS specification:

```

*)
ENTITY component_terminal_to_interconnect_module_interface_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
DERIVE
    usage_view_terminal : interconnect_module_interface_terminal :=
    SELF\component_feature_to_physical_usage_view_assignment.
    defined_part_feature;
    assigned_terminal : interconnect_or_assembly_module_select :=
    SELF\component_feature_to_physical_usage_view_assignment.
    assigned_design_object;
    UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_interconnect_module_interface_terminal_assignment

--tt tc1 wd9 modified
(*)

```

4.2.141 Component_terminal_to_interconnect_module_join_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_terminal_to_interconnect_module_join_terminal_assignment` is a type of `Component_feature_to_physical_usage_view_assignment` (see 4.2.123).

NOTE EXPRESS specification:

```

*)
ENTITY component_terminal_to_interconnect_module_join_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
DERIVE
  usage_view_terminal : interconnect_module_join_terminal :=
    SELF\component_feature_to_physical_usage_view_assignment.
    defined_part_feature;
    assigned_terminal : fabricated_feature_select :=
    SELF\component_feature_to_physical_usage_view_assignment.
    assigned_design_object;
  UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_interconnect_module_join_terminal_assignment

--tt tc1 wd9 modified
( *
```

4.2.142 Component_terminal_to_pca_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_terminal_to_pca_terminal_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Component_terminal_to_pca_terminal_assignment` is the relationship between a `Pca_terminal` (see 4.2.640) and one of an `Assembly_module_component_terminal` (see 4.2.45), a `Component_interface_terminal` (see 4.2.125), or an `Interconnect_module_component_terminal` (see 4.2.445), to provide traceability from the data in the design view to the data in the usage view for the design organization for archival purposes.

The data associated with a `Component_terminal_to_pca_terminal_assignment` are the following:

- `assigned_terminal`;
- `usage_view_terminal`.

4.2.142.1 assigned_terminal

The `assigned_terminal` specifies one of an `Assembly_module_component_terminal` (see 4.2.45), a `Component_interface_terminal` (see 4.2.125), or an `Interconnect_module_component_terminal` (see 4.2.445) for the `Component_terminal_to_pca_terminal_assignment`.

4.2.142.2 usage_view_terminal

The usage_view_terminal specifies the role of the Pca_terminal (see 4.2.640) for the Component_terminal_to_pca_terminal_assignment.

Constraints:

UR1: The usage_view_terminal shall be unique within a population of Component_terminal_to_pca_terminal_assignments.

NOTE EXPRESS specification:

```

*)
ENTITY component_terminal_to_pca_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
DERIVE
    usage_view_terminal : pca_terminal :=
      SELF\component_feature_to_physical_usage_view_assignment.
      defined_part_feature;
    assigned_terminal : pca_interface_select :=
      SELF\component_feature_to_physical_usage_view_assignment.
      assigned_design_object;
  UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_pca_terminal_assignment

( *
```

4.2.143 Component_termination_passage

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage is a type of Plated_passage (see 4.2.690) through which a Component_terminal (see 4.2.137) may pass. The properties of the Component_termination_passage are selected to support the Component_terminal.

NOTE 1 The model in this part of ISO 10303 is that the wall of the passage is directly supporting current flow.

NOTE 2 Support for the member of `Component_terminal` that passes through an `Inter_stratum_feature` (see 4.2.432) but is not assembled to the wall of that `Inter_stratum_feature` is provided by the `Cutout` (see 4.2.208) concept.

NOTE 3 Support for the member of `Component_terminal` that electrically is connected only to the surface of an `Interconnect_module` (see 4.2.440) is provided by the `Land` (see 4.2.490) concept.

NOTE 4 There is no requirement for the `Component_termination_passage` to completely penetrate the interconnect substrate.

EXAMPLE 1 Figure 15 is a detail of a layout drawing that superimposes components from the next level of assembly in order to illustrate the physical relationship of a `Component_termination_passage` to the terminals of those components. This example omits the details of how this part of ISO 10303 establishes the relationship between the terminal and the passage for clarity.

EXAMPLE 2 Figure 16 is a detail of a cross-section of an interconnect substrate showing a case of complete penetration of the substrate by the `Component_termination_passage`.

The data associated with a `Component_termination_passage` are the following:

- `access_mechanism`;
- `interface_access_mechanism`;
- `usage_definition`.

4.2.143.1 access_mechanism

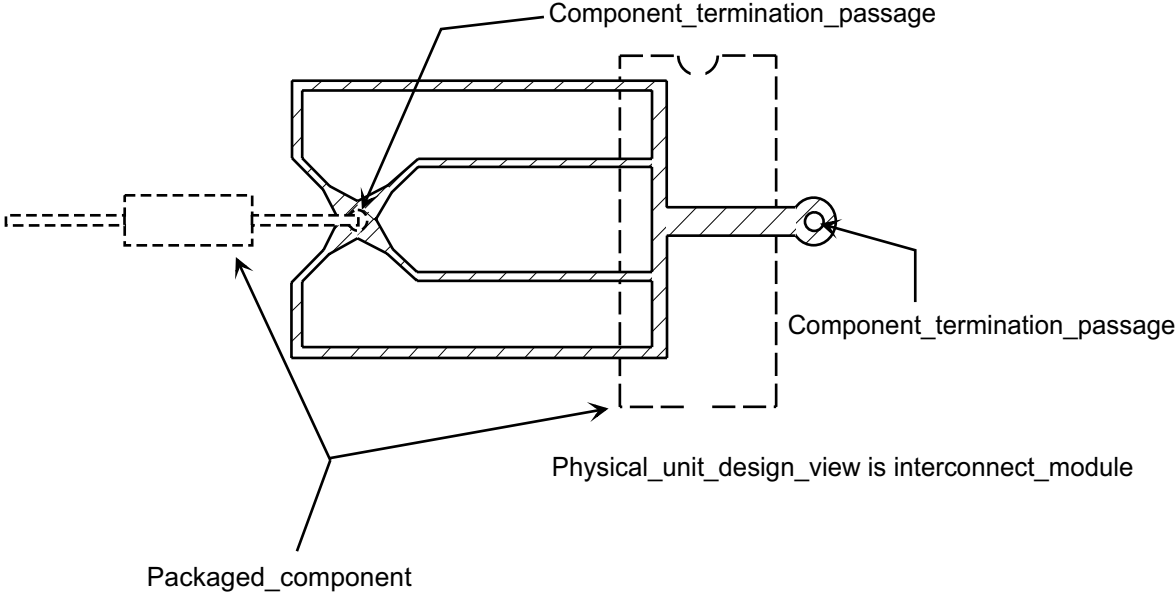
The `access_mechanism` is an inverse relationship that specifies that the existence of the `Component_termination_passage` is dependent on the existence of the `Component_termination_passage_join_terminal` (see 4.2.149) that specifies the `Component_termination_passage` as its associated_component. There shall be one or more `Component_termination_passage_join_terminal` for a `Component_termination_passage`.

4.2.143.2 interface_access_mechanism

The `interface_access_mechanism` is an inverse relationship that specifies that the existence of the `Component_termination_passage` is dependent on the existence of the `Component_termination_passage_interface_terminal` (see 4.2.148) that specifies the `Component_termination_passage` as its associated_component.

4.2.143.3 usage_definition

The `usage_definition` specifies the role of the `Component_termination_passage_template` (see 4.2.150) for the `Component_termination_passage`.



Note : Packaged_component are at next level of product structure.

Figure 15 – Component_termination_passage role in an assembly

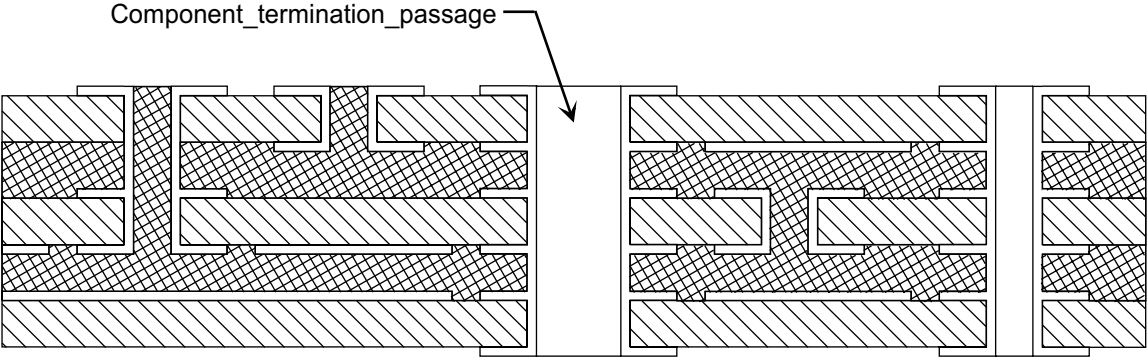


Figure 16 – Component_termination_passage cross-section view

Constraints:

WR1: A Component_termination_passage shall be a feature of size.

NOTE A Component_termination_passage is a feature of size, because it is part of the interface to another product in the assembly and the size of the passage shall be compatible with the appropriate assembly requirements of that product. Information about the other product's features needed for this function is specified in this part of ISO 10303 in the Default_component_termination_passage_definition specified by usage_definition.of_passage-technology.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage
  SUBTYPE OF (plated_passage);
  SELF\assembly_component.usage_definition :
  component_termination_passage_template;
  INVERSE
    access_mechanism :
    SET [1:?] OF component_termination_passage_join_terminal FOR
  associated_component;
  interface_access_mechanism :
  component_termination_passage_interface_terminal FOR
  associated_component;
END_ENTITY; -- component_termination_passage

( *
```

4.2.144 Component_termination_passage_and_contact_size_dependent_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_and_contact_size_dependent_land is a type of Land (see 4.2.490). The shape of the Component_termination_passage_and_contact_size_dependent_land is dependent both on the passage that is within the shape of the land (where that passage is included for the purpose of connecting the component terminal to a trace on the substrate) and on the size of the terminal that is going to be resting on the board. A Component_termination_passage_and_contact_size_dependent_land is an interface to a PCB or to a PWB, and therefore includes an external access mechanism.

NOTE A Component_termination_passage_and_contact_size_dependent_land principally provides a location for the solder joint.

The data associated with a `Component_termination_passage_and_contact_size_dependent_land` are the following:

- `interface_access_mechanism`;
- `reference_passage`;
- `usage_definition`.

4.2.144.1 `interface_access_mechanism`

The `interface_access_mechanism` is an inverse relationship that specifies that the existence of the `Component_termination_passage_and_contact_size_dependent_land` is dependent on the existence of the `Land_interface_terminal` (see 4.2.491) that specifies the `Component_termination_passage_and_contact_size_dependent_land` as its `associated_component`. This relationship is for external access only.

4.2.144.2 `reference_passage`

The `reference_passage` specifies the role of the `Component_termination_passage` (see 4.2.143) for the `Component_termination_passage_and_contact_size_dependent_land`.

4.2.144.3 `usage_definition`

The `usage_definition` specifies the role of the `Default_attachment_size_and_component_termination_passage_based_land_physical_template` (see 4.2.245) for the `Component_termination_passage_and_contact_size_dependent_land`.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_and_contact_size_dependent_land
  SUBTYPE OF (land);
  SELF\assembly_component.usage_definition :
  default_attachment_size_and_component_termination_passage_based_land_physical_template;
  reference_passage : component_termination_passage;
INVERSE
  interface_access_mechanism : land_interface_terminal FOR associated_component;
END_ENTITY; -- component_termination_passage_and_contact_size_dependent_land

( *
```

4.2.145

`Component_termination_passage_and_contact_size_dependent_non_functional_land`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_termination_passage_and_contact_size_dependent_non_functional_land` is a type of `Non_functional_land` (see 4.2.562). The shape of the `Component_termination_passage_and_contact_size_dependent_non_functional_land` is dependent both on the passage that is within the shape of the land (where that passage is included for the purpose of connecting the component terminal to a trace on the substrate) and on the size of the terminal that is going to be resting on the board.

NOTE A `Component_termination_passage_and_contact_size_dependent_non_functional_land` principally provides a location for the solder joint.

The data associated with a `Component_termination_passage_and_contact_size_dependent_non_functional_land` are the following:

- `interface_access_mechanism`;
- `reference_passage`;
- `usage_definition`.

4.2.145.1 interface_access_mechanism

The `interface_access_mechanism` is an inverse relationship that specifies that the existence of the `Component_termination_passage_and_contact_size_dependent_non_functional_land` is dependent on the existence of the `Non_functional_land_interface_terminal` (see ??) that specifies the `Component_termination_passage_and_contact_size_dependent_non_functional_land` as its `associated_component`. This relationship is for external access only.

4.2.145.2 reference_passage

The `reference_passage` specifies the role of the `Component_termination_passage` (see 4.2.143) for the `Component_termination_passage_and_contact_size_dependent_non_functional_land`.

4.2.145.3 usage_definition

The `usage_definition` specifies the role of the `Default_attachment_size_and_component_termination_passage_based_land_physical_template` (see 4.2.245) for the `Component_termination_passage_and_contact_size_dependent_non_functional_land`.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_and_contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  SELF\assembly_component.usage_definition :
  default_attachment_size_and_component_termination_passage_based_land_physical_template;
  reference_passage : component_termination_passage;
INVERSE
  interface_access_mechanism : non_functional_land_interface_terminal FOR associated_comp
END_ENTITY; -- component_termination_passage_and_contact_size_dependent_non_functional_land

( *

```

4.2.146 Component_termination_passage_dependent_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_dependent_land is a type of Land (see 4.2.490). The shape of the Component_termination_passage_dependent_land is dependent only on the passage that is within the shape of the land (where that passage is included for the purpose of connecting the component terminal to a trace on the substrate).

The data associated with a Component_termination_passage_dependent_land are the following:

- reference_passage;
- usage_definition.

4.2.146.1 reference_passage

The reference_passage specifies the role of the Component_termination_passage (see 4.2.143) for the Component_termination_passage_dependent_land.

4.2.146.2 usage_definition

The usage_definition specifies the role of the Default_component_termination_passage_based_land_physical_template (see 4.2.248) for the Component_termination_passage_dependent_land.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_dependent_land
  SUBTYPE OF (land);
    reference_passage                : component_termination_passage;
    SELF\assembly_component.usage_definition :
default_component_termination_passage_based_land_physical_template;
END_ENTITY; -- component_termination_passage_dependent_land

( *

```

4.2.147 Component_termination_passage_dependent_non_functional_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_dependent_non_functional_land is a type of Non_functional_land (see 4.2.562). The shape of the Component_termination_passage_dependent_non_functional_land is dependent only on the passage that is within the shape of the land (where that passage is included for the purpose of connecting the component terminal to a trace on the substrate).

The data associated with a Component_termination_passage_dependent_non_functional_land are the following:

- reference_passage;
- usage_definition.

4.2.147.1 reference_passage

The reference_passage specifies the role of the Component_termination_passage (see 4.2.143) for the Component_termination_passage_dependent_non_functional_land.

4.2.147.2 usage_definition

The usage_definition specifies the role of the Default_component_termination_passage_based_land_physical_template (see 4.2.248) for the Component_termination_passage_dependent_non_functional_land.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
    reference_passage                : component_termination_passage;
    SELF\assembly_component.usage_definition :
default_component_termination_passage_based_land_physical_template;
END_ENTITY; -- component_termination_passage_dependent_non_functional_land

( *

```

4.2.148 Component_termination_passage_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_interface_terminal is a type of Component_feature (see 4.2.118). The Component_termination_passage_interface_terminal is the internal wall of the passage only and is the complete internal wall. It does not extend to the exterior top or bottom edges of the passage. It does not include an exterior surface of the passage. This terminal is the definition of an interface mechanism for an interconnect.

NOTE In normal usage, the interior wall has a smaller diameter than the exterior wall.

The data associated with a Component_termination_passage_interface_terminal are the following:

- associated_component;
- definition.

4.2.148.1 associated_component

The associated_component specifies the role of the Component_termination_passage (see 4.2.143) for the Component_termination_passage_interface_terminal.

4.2.148.2 definition

The definition specifies the role of the Component_termination_passage_template_interface_terminal (see 4.2.151) for the Component_termination_passage_interface_terminal.

EXAMPLE 1 One use of this capability occurs when a member of Component_feature_to_physical_usage-view_assignment references a member of Component_termination_passage_interface_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_interface_terminal
  SUBTYPE OF (laminate_component_interface_terminal);
  SELF\component_feature.definition :
component_termination_passage_template_interface_terminal;
  SELF\component_feature.associated_component : component_termination_passage;
END_ENTITY; -- component_termination_passage_interface_terminal

( *

```

4.2.149 Component_termination_passage_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_join_terminal is a type of Component_terminal (see 4.2.137). The Component_termination_passage_join_terminal is the exterior surface of the passage. The exterior surface has a larger nominal diameter than the interior surface.

NOTE One Component_termination_passage_join_terminal may be related to multiple Land_join_terminals, since each stratum may have a Land (see 4.2.490) associated with the Component_termination_passage (see 4.2.143).

The data associated with a Component_termination_passage_join_terminal are the following:

- associated_component;
- definition.

4.2.149.1 associated_component

The associated_component specifies the role of the Component_termination_passage for the Component_termination_passage_join_terminal.

4.2.149.2 definition

The definition specifies the role of the Component_termination_passage_template_join_terminal (see 4.2.152) for the Component_termination_passage_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_join_terminal
  SUBTYPE OF (laminate_component_join_terminal);
  SELF\component_feature.definition      :
component_termination_passage_template_join_terminal;
  SELF\component_feature.associated_component : component_termination_passage;
END_ENTITY; -- component_termination_passage_join_terminal

( *

```

4.2.150 Component_termination_passage_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_template is a type of Inter_stratum_feature_template (see 4.2.435). The Component_termination_passage_template defines all shape data for the associated Component_termination_passage (for which it is the usage_definition), and also where the connection (dependency) on passage technology is established for the properties.

NOTE Properties other than shape are to be found in the of_passage_technology.

The data associated with a Component_termination_passage_template are the following:

- access_mechanisms;
- of_passage_technology.

4.2.150.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Component_termination_passage_template is dependent on the existence of the Component_termination_passage_template_terminal (see 4.2.153) that specifies the Component_termination_passage_template as its associated_definition. There shall be two or more Component_termination_passage_template_terminals for a Component_termination_passage_template.

4.2.150.2 of_passage_technology

The of_passage_technology specifies the role of the Default_component_termination_passage_definition (see 4.2.249) for the Component_termination_passage_template.

Constraints:

WR1: One access_mechanisms shall be a Component_termination_passage_template_interface_terminal.

WR2: One access_mechanisms shall be a Component_termination_passage_template_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_template
  SUBTYPE OF (inter_stratum_feature_template);
  SELF\inter_stratum_feature_template.of_passage_technology :
default_component_termination_passage_definition;
  INVERSE
    access_mechanisms : SET [2:?] OF component_termination_passage_template_terminal
      FOR associated_definition;
END_ENTITY; -- component_termination_passage_template

( *
```

4.2.151 Component_termination_passage_template_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Component_termination_passage_template_interface_terminal is a type of Component_termination_passage_template_terminal (see 4.2.153), an occurrence of which provides an interface capability to the next level of assembly.

NOTE Use of this capability occurs when a member of Component_feature_to_physical_usage_view_assignment references a member of an occurrence of Component_termination_passage_template_interface_terminal.

EXAMPLE 1 Figure 17 illustrates details related to the definition (in a usage view) and occurrence (in a design view) of the terminals associated with a Component_termination_passage_template.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_template_interface_terminal
  SUBTYPE OF (component_termination_passage_template_terminal);
END_ENTITY; -- component_termination_passage_template_interface_terminal

( *
```

4.2.152 Component_termination_passage_template_join_terminal

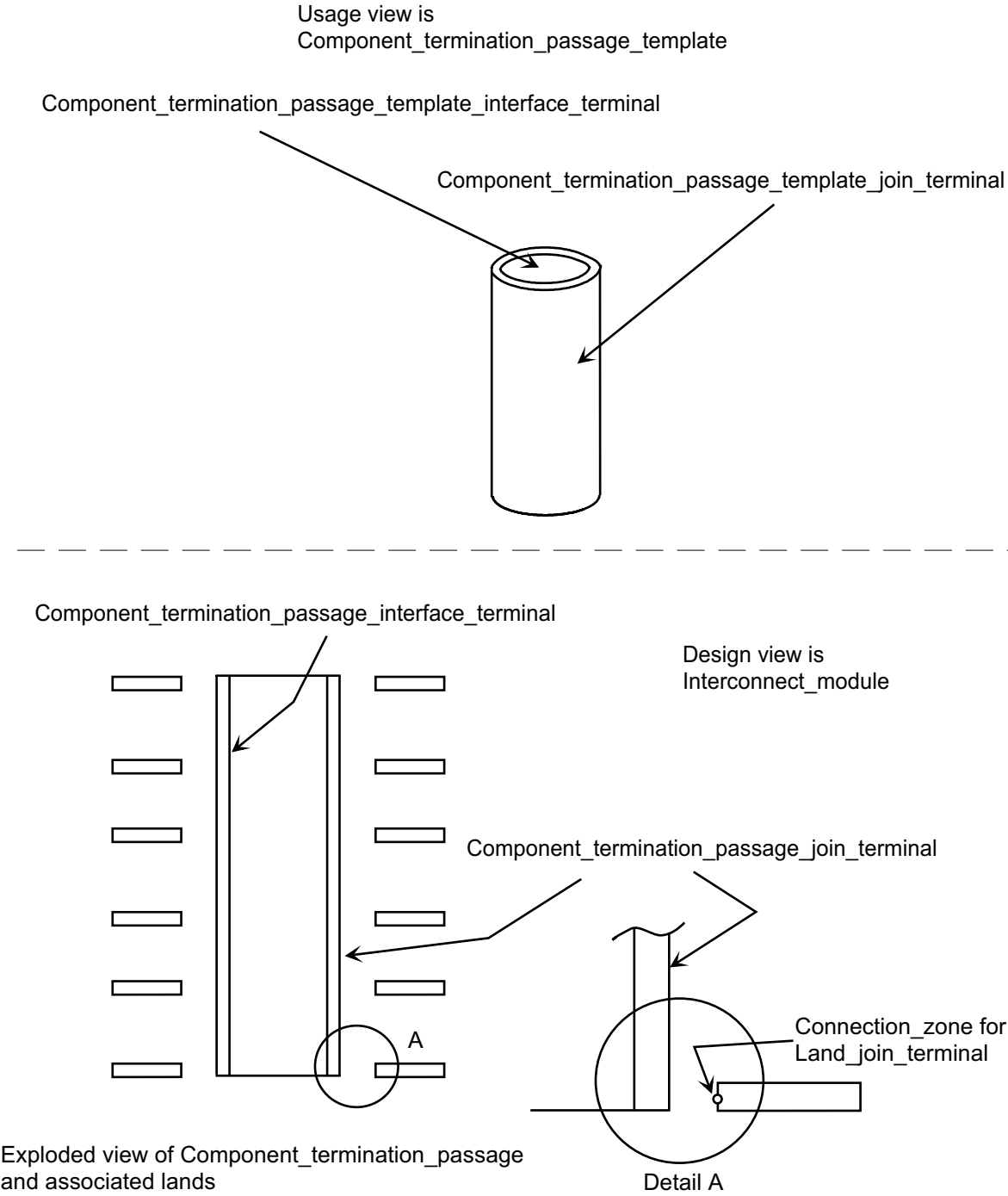


Figure 17 – Component_termination_passage_template terminals

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_termination_passage_template_join_terminal` is a type of `Component_termination_passage_template_terminal` (see 4.2.153), an occurrence of which provides a fabrication join capability, within an optionally constrained set of `Stratum` (see 4.2.845).

The data associated with a `Component_termination_passage_template_join_terminal` are the following:

— `disallowed_inter_stratum_extent`.

4.2.152.1 `disallowed_inter_stratum_extent`

The `disallowed_inter_stratum_extent` specifies the `Inter_stratum_extent` (see 4.2.431) for which a connection to the `Component_termination_passage_template_join_terminal` is disallowed. The `disallowed_inter_stratum_extent` need not be supplied for a `Component_termination_passage_template_join_terminal`. There may be more than one `Inter_stratum_extent` for a `Component_termination_passage_template_join_terminal`.

NOTE EXPRESS specification:

```
*)
ENTITY component_termination_passage_template_join_terminal
  SUBTYPE OF (component_termination_passage_template_terminal);
  --tt tc1 wd8 modified
    disallowed_inter_stratum_extent : OPTIONAL SET [1:?] OF inter_stratum_extent;
END_ENTITY; -- component_termination_passage_template_join_terminal

( *
```

4.2.153 `Component_termination_passage_template_terminal`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Component_termination_passage_template_terminal` is a type of `Shape_element` (see 4.2.810). A `Component_termination_passage_template_terminal` is one of a `Component_termination_passage_template_interface_terminal` (see 4.2.151) or a `Component_termination_passage_template_join_terminal` (see 4.2.152).

The data associated with a `Component_termination_passage_template_terminal` are the following:

— `associated_definition`;

— `name`;

— `terminal_connection_zone`.

4.2.153.1 associated_definition

The associated_definition specifies the role of the Component_termination_passage_template (see 4.2.150) for the Component_termination_passage_template_terminal.

4.2.153.2 name

The name specifies the string identifying the Component_termination_passage_template_terminal.

4.2.153.3 terminal_connection_zone

The terminal_connection_zone specifies the role of the Connection_zone (see 4.2.185) for the Component_termination_passage_template_terminal. The terminal_connection_zone is optional and need not be supplied if the data is unavailable in the source system. If the data is available in the source system, it shall be supplied.

Constraints:

UR1: The combination of the associated_definition and the name shall be unique within a population of Component_termination_passage_template_terminals.

WR1: The points comprising the terminal_connection_zones shall be defined in the coordinate system of the device being accessed via the terminal_connection_zones.

NOTE EXPRESS specification:

```

*)
ENTITY component_termination_passage_template_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (
    component_termination_passage_template_interface_terminal,
    component_termination_passage_template_join_terminal))
  SUBTYPE OF (shape_element);
  associated_definition      : component_termination_passage_template;
  terminal_connection_zone  : OPTIONAL SET [1:?] OF connection_zone;
  SELF\shape_element.name  : ee_name;
INVERSE
  terminal_shape            : SET [0:1] OF usage_concept_usage_relationship FOR
  associated_usage;
  UNIQUE
  UR1 : name, associated_definition;
END_ENTITY; -- component_termination_passage_template_terminal

( *
```

4.2.154 Compound_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Compound_characteristic is a type of Characteristic (see 4.2.98). A Compound_characteristic may be a Characteristic_data_table (see 4.2.100). A Compound_characteristic is a property of an item. A Compound_characteristic helps to compose a structured characteristic, in conjunction with Compound_characteristic_definition (see ??).

NOTE A Compound_characteristic may describe either a physical item, like a realized part, or a description that is unrealizable, such as a simulation model. A Compound_characteristic is most often used to describe a table of values that may range over a discrete parameter, e.g., a table of response values vs. frequency or time.

The data associated with a Compound_characteristic are the following:

— property_value.

4.2.154.1 property_value

The property_value specifies the role of a Compound_characteristic_definition (see ??) for the Compound_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY compound_characteristic
  ABSTRACT SUPERTYPE OF (ONEOF (characteristic_data_set, characteristic_data_table))
  SUBTYPE OF (characteristic);
  SELF\characteristic.property_value : compound_characteristic_definition;
END_ENTITY; -- compound_characteristic
(*
```

4.2.155 Composite_curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_curve is a type of Curve (see 4.2.203) that combines more than one Curve into its structure. The Composite_curve shall be continuous.

The data associated with a Composite_curve are the following:

— curve_segment.

4.2.155.1 curve_segment

The curve_segment specifies the role of the Composite_curve_segment (see 4.2.156) for the Composite_curve. There shall be one or more Composite_curve_segment for a Composite_curve. No Composite_curve_segment shall be specified more than once as a curve_segment for a particular Composite_curve.

NOTE EXPRESS specification:

```
*)
ENTITY composite_curve
  SUBTYPE OF (curve);
  curve_segment : LIST [1:?] OF UNIQUE composite_curve_segment;
END_ENTITY; -- composite_curve
```

(*

4.2.156 Composite_curve_segment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_curve_segment is a type of Managed_design_object (see 4.2.524) that is a segment of a Composite_curve (see 4.2.155). A Composite_curve_segment associates Curves with a Composite_curve to help compose the Composite_curve.

The data associated with a Composite_curve_segment are the following:

- segment_curve;
- using_curves.

4.2.156.1 segment_curve

The segment_curve specifies the role of the Curve (see 4.2.203) for the Composite_curve_segment.

4.2.156.2 using_curves

The using_curves is an inverse relationship that specifies that the existence of the Composite_curve_segment is dependent on the existence of the Composite_curve (see 4.2.155) that specifies the Composite_curve_segment as its curve_segment. There shall be one or more Composite_curve for a Composite_curve_segment.

NOTE EXPRESS specification:

```
*)
ENTITY composite_curve_segment
  SUBTYPE OF (managed_design_object);
  segment_curve : curve;
  INVERSE
  using_curves : SET [1:?] OF composite_curve FOR curve_segment;
END_ENTITY; -- composite_curve_segment
```

(*

4.2.157 Composite_feature_locating_position_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_feature_locating_position_tolerance is a type of Position_tolerance_with_referenced_datum_system (see 4.2.696). The Composite_feature_locating_position_tolerance is used in conjunction with one to three Composite_feature_relating_position_tolerance (see 4.2.158) to apply a less restrictive specification on the location of a Shape_element (see 4.2.810) group with respect to one or more Datum (see 4.2.214) than on the location and orientation of those Shape_element within the group with respect to each other. The Tolerance_zone (see 4.2.913) defined by the Composite_feature_locating_position_tolerance is located relative to the Datum_system (see 4.2.236) the definition of which is specified as the datum_system_definition.

NOTE 1 The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

NOTE 2 Composite positional tolerancing is discussed in 5.4.1 of ASME Y14.5M.

Constraints:

WR1: The Composite_feature_locating_position_tolerance shall not also be a Composite_feature_relating_position_tolerance.

WR2: The Composite_feature_locating_position_tolerance shall be specified as the refined_control by exactly one Composite_feature_relating_position_tolerance.

WR3: The Shape_element that is specified as the toleranced_shape_element shall be a Composite_group_shape_element.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

NOTE EXPRESS specification:

```
*)
ENTITY composite_feature_locating_position_tolerance
  SUBTYPE OF (position_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_feature_locating_position_tolerance

( *
```

4.2.158 Composite_feature_relating_position_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_feature_relating_position_tolerance` is a type of `Position_tolerance` (see 4.2.695) that is used in conjunction with a `Composite_feature_locating_position_tolerance` (see 4.2.157) and possibly two other `Composite_feature_relating_position_tolerance` to apply a less restrictive specification on the location of a `Shape_element` (see 4.2.810) group with respect to one or more `Datum` (see 4.2.214) than on the location and orientation of those `Shape_element` within the group with respect to each other. If the `Composite_feature_relating_position_tolerance` is also a `Position_tolerance_with_referenced_datum_system` (see 4.2.696), the defined `Tolerance_zone` (see 4.2.913) is oriented but not located relative to the `Datum_system` (see 4.2.236) the definition of which is specified as the `datum_system_definition`.

The data associated with a `Composite_feature_relating_position_tolerance` are the following:

— `refined_control`.

NOTE The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` entity of which this entity is a subtype.

4.2.158.1 `refined_control`

The `refined_control` specifies either a `Composite_feature_locating_position_tolerance` (see 4.2.157) or another `Composite_feature_relating_position_tolerance` to which this `Composite_feature_relating_position_tolerance` assists in specifying design requirements by specifying a further refinement of the location and orientation of the group of `Shape_element` (see 4.2.810) in relation to each other.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` by this `Composite_feature_relating_position_tolerance` shall be the same instance of `Shape_element` as specified as the `toleranced_shape_element` by the `Position_tolerance` that is specified as the `refined_control`.

WR2: If this `Composite_feature_relating_position_tolerance` is also a `Position_tolerance_without_referenced_datum_system`, it shall not be specified as a `refined_control` by another `Composite_feature_relating_position_tolerance`.

WR3: If this `Composite_feature_relating_position_tolerance` specifies a `Composite_feature_relating_position_tolerance` as the `refined_control`, this `Composite_feature_relating_position_tolerance` shall either be a `Position_tolerance_without_referenced_datum_system` or have less `Datum` in its `Datum_system` than the `Datum_system` of the `Composite_feature_relating_position_tolerance` specified as the `refined_control`.

WR4: If this `Composite_feature_relating_position_tolerance` specifies a `Composite_feature_locating_position_tolerance` as the `refined_control`, this `Composite_feature_relating_position_tolerance` shall either be a `Position_tolerance_without_referenced_datum_system` or have less than or an equal number of `Datum` in its `Datum_system` than the `Datum_system` of the `Composite_feature_locating_position_tolerance` specified as the `refined_control`.

WR5: If this `Composite_feature_relating_position_tolerance` is a `Position_tolerance_with_referenced_datum_system`, the `Datum` in its `Datum_system` shall be in the `Datum_system` of the `Position_tolerance` specified as the `refined_control` and shall be assigned the same datum precedence.

WR6: If this `Composite_feature_relating_position_tolerance` specifies a `Position_tolerance` that specifies a `Datum_system_defining_conditions_without_material_conditions` as the `datum_system_definition`, the `Position_tolerance` specified as the `refined_control` shall also specify a `Datum_system_defining_conditions_without_material_conditions` as its `datum_system_definition`.

WR7: If this `Composite_feature_relating_position_tolerance` specifies a `Position_tolerance` that specifies a `Datum_system_defining_conditions_with_material_conditions` as the `datum_system_definition`; the `Position_tolerance` specified as the `refined_control` shall also specify a `Datum_system_defining_conditions_with_material_conditions` as its `datum_system_definition`, and each `Datum_feature` of the `Datum_system` of this `Composite_feature_relating_position_tolerance` shall be assigned the same material condition property as is assigned to it in the `Datum_system` of the `Position_tolerance` specified as the `refined_control`.

WR8: The size of the `Tolerance_zone_boundary` of this `Composite_feature_relating_position_tolerance` shall be less than the size of the `Tolerance_zone_boundary` of the `Position_tolerance` specified as the `refined_control`.

WR9: Both the `Composite_feature_relating_position_tolerance` and the `Position_tolerance` specified as the `refined_control` shall specify the same type of `Tolerance_zone_definition`.

WR10: The `Composite_feature_relating_position_tolerance` shall be specified as a `refined_control` by at most one `Composite_feature_relating_position_tolerance`.

WR11: Both the `Composite_feature_relating_position_tolerance` and the `Position_tolerance` specified as the `refined_control` shall specify the same type of `Tolerance_zone_boundary_definition`.

WR12: Both the `Composite_feature_relating_position_tolerance` and the `Position_tolerance` specified as the `refined_control` shall specify the same type of `Boundary_size_characteristic`.

NOTE EXPRESS specification:

```
*)
ENTITY composite_feature_relating_position_tolerance
  SUBTYPE OF (position_tolerance);
    refined_control : locating_or_relating_position_tolerance;
END_ENTITY; -- composite_feature_relating_position_tolerance

( *
```

4.2.159 Composite_group_shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_group_shape_element` is a type of `Composite_shape_element` (see 4.2.162) in which the composite may be thought of as a group of individual `Shape_element` (see 4.2.810) in which any property that is applied to the `Composite_group_shape_element` is applied to each of the constituent `Shape_elements` on an individual basis.

EXAMPLE 1 A group of cylindrical holes in a printed circuit board may be grouped with a `Composite_group_shape_element` for the purpose of applying the same `Cylindricity_tolerance` to each of the constituent holes in the group.

Constraints:

WR1: If all the `Shape_elements` that compose the `Composite_group_shape_element` are of the same type then the `Composite_shape_element` shall also be of that type.

EXAMPLE 2 If all the `Shape_elements` that compose a `Composite_group_shape_element` are of type `Centre_axis` then, the `Composite_group_shape_element` will also be of type `Centre_axis`.

NOTE EXPRESS specification:

```
* )
ENTITY composite_group_shape_element
  SUBTYPE OF (composite_shape_element);
END_ENTITY; -- composite_group_shape_element

( *
```

4.2.160 `Composite_linear_profile_locating_tolerance`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_linear_profile_locating_tolerance` is a type of `Profile_of_any_line_tolerance_with_referenced_datum_system` (see 4.2.745). The `Composite_linear_profile_locating_tolerance` is used in conjunction with one to three `Composite_linear_profile_refinement_tolerance` (see 4.2.161) to apply a more restrictive specification on the form, orientation, and possibly size of a `Physical_feature_or_part_template` (see 4.2.658) than on the location of the `Physical_feature_or_part_template`. The `Tolerance_zone` (see 4.2.913) defined by the `Composite_linear_profile_locating_tolerance` is located relative to the `Datum_system` (see 4.2.236) the definition of which is specified as the `datum_system_definition`.

NOTE The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` entity of which this entity is a subtype.

Constraints:

WR1: The `Composite_linear_profile_locating_tolerance` shall not also be a `Composite_linear_profile_refinement_tolerance`.

WR2: The `Composite_linear_profile_locating_tolerance` shall be specified as the `refined_control` by exactly one `Composite_linear_profile_refinement_tolerance`.

NOTE EXPRESS specification:

```
*)
ENTITY composite_linear_profile_locating_tolerance
  SUBTYPE OF (profile_of_any_line_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_linear_profile_locating_tolerance

( *
```

4.2.161 Composite_linear_profile_refinement_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_linear_profile_refinement_tolerance` is a type of `Profile_of_any_line_tolerance` (see 4.2.744) that is used in conjunction with a `Composite_linear_profile_locating_tolerance` (see 4.2.160) and possibly two other `Composite_linear_profile_refinement_tolerance` to apply a more restrictive specification on the form, orientation, and possibly size of a `Physical_feature_or_part_template` (see 4.2.658) than on the location of the `Physical_feature_or_part_template`. If the `Composite_linear_profile_refinement_tolerance` is also a `Profile_of_any_line_tolerance_with_referenced_datum_system` (see 4.2.745) the defined `Tolerance_zone` (see 4.2.913) is oriented but not located relative to the `Datum_system` (see 4.2.236) the definition of which is specified as the `datum_system_definition`.

NOTE The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422) entity of which this entity is a subtype.

The data associated with a `Composite_linear_profile_refinement_tolerance` are the following:

— `refined_control`.

4.2.161.1 refined_control

The `refined_control` specifies either a `Composite_linear_profile_locating_tolerance` (see 4.2.160) or another `Composite_linear_profile_refinement_tolerance` to which this `Composite_linear_profile_refinement_tolerance` assists in specifying design requirements by specifying a further refinement of the form, orientation, and possibly size of the `Physical_feature_or_part_template` (see 4.2.658) specified as the `toleranced_shape_element`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

Constraints:

WR1: The `Physical_feature_or_part_template` specified as the `toleranced_shape_element` by this `Composite_linear_profile_refinement_tolerance` shall be the same instance of `Physical_feature_or_part_template` as specified as the `toleranced_shape_element` by the `Profile_of_any_line_tolerance` that is specified as the `refined_control`.

WR2: If this `Composite_linear_profile_refinement_tolerance` is also a `Profile_of_any_line_tolerance_without_referenced_datum_system`, it shall not be specified as a `refined_control` by another `Composite_linear_profile_refinement_tolerance`.

WR3: If this `Composite_linear_profile_refinement_tolerance` specifies a `Composite_linear_profile_refinement_tolerance` as the `refined_control`, this `Composite_linear_profile_refinement_tolerance` shall either be a `Profile_of_any_line_tolerance_without_referenced_datum_system` or have less `Datum` in its `Datum_system` than the `Datum_system` of the `Composite_linear_profile_refinement_tolerance` specified as the `refined_control`.

WR4: If this `Composite_linear_profile_refinement_tolerance` specifies a `Composite_linear_profile_locating_tolerance` as the `refined_control`, this `Composite_linear_profile_refinement_tolerance` shall either be a `Profile_of_any_line_tolerance_without_referenced_datum_system` or have less than or an equal number of `Datum` in its `Datum_system` than the `Datum_system` of the `Composite_linear_profile_locating_tolerance` specified as the `refined_control`.

WR5: If this `Composite_linear_profile_refinement_tolerance` is a `Profile_of_any_line_tolerance_with_referenced_datum_system`, the `Datum` in its `Datum_system` shall be in the `Datum_system` of the `Profile_of_any_line_tolerance` specified as the `refined_control` and shall be assigned the same datum precedence.

WR6: If this `Composite_linear_profile_refinement_tolerance` specifies a `Profile_of_any_line_tolerance` that specifies a `Datum_system_defining_conditions_without_material_conditions` as the `datum_system_definition`, the `Profile_of_any_line_tolerance` specified as the `refined_control` shall also specify a `Datum_system_defining_conditions_without_material_conditions` as its `datum_system_definition`.

WR7: If this `Composite_linear_profile_refinement_tolerance` specifies a `Profile_of_any_line_tolerance` that specifies a `Datum_system_defining_conditions_with_material_conditions` as the `datum_system_definition`; the `Profile_of_any_line_tolerance` specified as the `refined_control` shall also specify a `Datum_system_defining_conditions_with_material_conditions` as its `datum_system_definition`, and each `Datum_feature` of the `Datum_system` of this `Composite_linear_profile_refinement_tolerance` shall be assigned the same material condition property as is assigned to it in the `Datum_system` of the `Profile_of_any_line_tolerance` specified as the `refined_control`.

WR8: The size of the `Tolerance_zone_boundary` of this `Composite_linear_profile_refinement_tolerance` shall be less than the size of the `Tolerance_zone_boundary` of the `Profile_of_any_line_tolerance` specified as the `refined_control`.

WR9: The `Tolerance_zone_boundary` of this `Composite_linear_profile_refinement_tolerance` shall be based on the same `True_profile` as the `Tolerance_zone_boundary` of the `Profile_of_any_line_tolerance` specified as the `refined_control`.

WR10: The `Composite_linear_profile_refinement_tolerance` shall be specified as a `refined_control` by at most one `Composite_linear_profile_refinement_tolerance`.

WR11: If the `Tolerance_zone_definition` that is specified as the `tolerance_zone_or_boundary_definition` is a `Tolerance_zone_definition_with_specified_orientation`, the `Tolerance_zone_definition` that is specified as the `tolerance_zone_or_boundary_definition` by the `Profile_of_any_line_tolerance` that is specified as the `refined_control` shall also be a `Tolerance_zone_definition_with_specified_orientation`, and the orientations shall be the same.

NOTE EXPRESS specification:

```
*)
ENTITY composite_linear_profile_refinement_tolerance
  SUBTYPE OF (profile_of_any_line_tolerance);
    refined_control : locating_or_refinement_linear_profile_tolerance;
END_ENTITY; -- composite_linear_profile_refinement_tolerance

( *
```

4.2.162 Composite_shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

`Composite_shape_element` is a type of `Shape_element` (see 4.2.810) that is composed of two or more constituent `Shape_elements`. A `Composite_shape_element` is either a `Composite_group_shape_element` (see 4.2.159) or a `Composite_unit_shape_element` (see 4.2.167).

The data associated with a `Composite_shape_element` are the following:

— `composing_relationships`.

4.2.162.1 composing_relationships

The `composing_relationships` is an inverse relationship that specifies that the existence of the `Composite_shape_element` is dependent on the existence of the `Shape_element_composing_relationship` (see 4.2.812) that specifies the `Composite_shape_element` as its `relating_shape_element`. There shall be two or more `Shape_element_composing_relationship` for a `Composite_shape_element`.

NOTE EXPRESS specification:

```

*)
ENTITY composite_shape_element
  ABSTRACT SUPERTYPE OF (ONEOF (composite_group_shape_element,
    composite_unit_shape_element))
  SUBTYPE OF (shape_element);
  INVERSE
    composing_relationships : SET [2:?] OF shape_element_composing_relationship FOR
      relating_shape_element;
END_ENTITY; -- composite_shape_element

( *

```

4.2.163 Composite_signal_property

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_signal_property is a type of Signal_property (see 4.2.824) that combines more than one signal property into its structure. This part of ISO 10303 does not predefine the domain semantics of the structural relationship.

The data associated with a Composite_signal_property are the following:

— signal_property_structure.

4.2.163.1 signal_property_structure

The signal_property_structure is an inverse relationship that specifies that the existence of the Composite_signal_property is dependent on the existence of the Composite_signal_property_relationship (see 4.2.164) that specifies the Composite_signal_property as its composition. There shall be one or more Composite_signal_property_relationship for a Composite_signal_property.

NOTE EXPRESS specification:

```

*)
ENTITY composite_signal_property
  SUBTYPE OF (signal_property);
  INVERSE
    signal_property_structure : SET [1:?] OF composite_signal_property_relationship FOR
      composition;
END_ENTITY; -- composite_signal_property

( *

```

4.2.164 Composite_signal_property_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_signal_property_relationship is a type of Managed_design_object (see 4.2.524). A Composite_signal_property_relationship is the aggregation of the Signal_property (see 4.2.824) that compose the Composite_signal_property (see 4.2.163).

The data associated with a Composite_signal_property_relationship are the following:

- aggregation_operation;
- composition;
- signal_property_structural_element.

4.2.164.1 aggregation_operation

The aggregation_operation specifies the External_definition (see 4.2.361) for the mathematical model for the inclusion of an element in the Composite_signal_property_relationship. The structure that the aggregation_operation helps to implement shall be specified by an ISO or an IEC standard.

4.2.164.2 composition

The composition specifies the role of the Composite_signal_property (see 4.2.163) for the Composite_signal_property_relationship. The structure shall be defined by an ISO or IEC standard.

4.2.164.3 signal_property_structural_element

The signal_property_structural_element specifies the Signal_property (see 4.2.824) that is being included in the Composite_signal_property_relationship.

NOTE Signal property composition may be domain specific and is covered by available standards.

NOTE EXPRESS specification:

```
*)
ENTITY composite_signal_property_relationship
  SUBTYPE OF (managed_design_object);
    composition                : composite_signal_property;
    signal_property_structural_element : signal_property;
    aggregation_operation       : external_definition;
END_ENTITY; -- composite_signal_property_relationship

( *
```

4.2.165 Composite_surface_profile_locating_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_surface_profile_locating_tolerance` is a type of `Profile_of_any_surface_tolerance_with_referenced_datum_system` (see 4.2.748). The `Composite_surface_profile_locating_tolerance` is used in conjunction with one to three `Composite_surface_profile_refinement_tolerance` (see 4.2.166) to apply a more restrictive specification on the form, orientation, and possibly size of a `Physical_feature_or_part_template` (see 4.2.658) than on the location of the `Physical_feature_or_part_template`. The `Tolerance_zone` (see 4.2.913) defined by the `Composite_surface_profile_locating_tolerance` is located relative to the `Datum_system` (see 4.2.236) the definition of which is specified as the `datum_system_definition`.

NOTE 1 The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` entity of which this entity is a subtype.

NOTE 2 Composite profile tolerancing is discussed in 6.5.9.1 of ASME Y14.5M.

Constraints:

WR1: The `Composite_surface_profile_locating_tolerance` shall not also be a `Composite_surface_profile_refinement_tolerance`.

WR2: The `Composite_surface_profile_locating_tolerance` shall be specified as the `refined_control` by exactly one `Composite_surface_profile_refinement_tolerance`.

NOTE EXPRESS specification:

```
* )
ENTITY composite_surface_profile_locating_tolerance
  SUBTYPE OF (
    profile_of_any_surface_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_surface_profile_locating_tolerance

( *
```

4.2.166 Composite_surface_profile_refinement_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Composite_surface_profile_refinement_tolerance` is a type of `Profile_of_any_surface_tolerance` (see 4.2.747) that is used in conjunction with a `Composite_surface_profile_locating_tolerance` (see 4.2.165) and possibly two other `Composite_surface_profile_refinement_tolerance` to apply a more restrictive specification on the form, orientation, and possibly size of a `Physical_feature_or_part_template` (see 4.2.658) than on the location of the `Physical_feature_or_part_template`. If the `Composite_surface_profile_refinement_tolerance` is also a `Profile_of_any_surface_tolerance_with_referenced_datum_system` (see 4.2.748) the defined `Tolerance_zone` (see 4.2.913) is oriented but not located relative to the `Datum_system` (see 4.2.236) the definition of which is specified as the `datum_system_definition`.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system (see 4.2.422) entity of which this entity is a subtype.

The data associated with a Composite_surface_profile_refinement_tolerance are the following:

— refined_control.

4.2.166.1 refined_control

The refined_control specifies either a Composite_surface_profile_locating_tolerance (see 4.2.165) or another Composite_surface_profile_refinement_tolerance to which this Composite_surface_profile_refinement_tolerance assists in specifying design requirements by specifying a further refinement of the form, orientation, and possibly size of the Physical_feature_or_part_template (see 4.2.658) specified as the toleranced_shape_element.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

Constraints:

WR1: The Physical_feature_or_part_template specified as the toleranced_shape_element by this Composite_surface_profile_refinement_tolerance shall be the same instance of Physical_feature_or_part_template as specified as the toleranced_shape_element by the Profile_of_any_surface_tolerance that is specified as the refined_control.

WR2: If this Composite_surface_profile_refinement_tolerance is also a Profile_of_any_surface_tolerance_without_referenced_datum_system, it shall not be specified as a refined_control by another Composite_surface_profile_refinement_tolerance.

WR3: If this Composite_surface_profile_refinement_tolerance specifies a Composite_surface_profile_refinement_tolerance as the refined_control, this Composite_surface_profile_refinement_tolerance shall either be a Profile_of_any_surface_tolerance_without_referenced_datum_system or have less Datum in its Datum_system than the Datum_system of the Composite_surface_profile_refinement_tolerance specified as the refined_control.

WR4: If this Composite_surface_profile_refinement_tolerance specifies a Composite_surface_profile_locating_tolerance as the refined_control, this Composite_surface_profile_refinement_tolerance shall either be a Profile_of_any_surface_tolerance_without_referenced_datum_system or have less than or an equal number of Datum in its Datum_system than the Datum_system of the Composite_surface_profile_locating_tolerance specified as the refined_control.

WR5: If this Composite_surface_profile_refinement_tolerance is a Profile_of_any_surface_tolerance_with_referenced_datum_system, the Datum in its Datum_system shall be in the Datum_system of the Profile_of_any_surface_tolerance specified as the refined_control and shall be assigned the same datum precedence.

WR6: If this Composite_surface_profile_refinement_tolerance specifies a Profile_of_any_surface_tolerance that specifies a Datum_system_defining_conditions_without_material_conditions as the datum_system_definition, the Profile_of_any_surface_tolerance specified as the refined_control shall also specify a Datum_system_defining_conditions_without_material_conditions as its datum_system_definition.

WR7: If this Composite_surface_profile_refinement_tolerance specifies a Profile_of_any_surface_tolerance that specifies a Datum_system_defining_conditions_with_material_conditions as the datum_system_definition; the Profile_of_any_surface_tolerance specified as the refined_control shall also specify a Datum_system_defining_conditions_with_material_conditions as its datum_system_definition, and each Datum_feature of the Datum_system of this Composite_surface_profile_refinement_tolerance shall be assigned the same material condition property as is assigned to it in the Datum_system of the Profile_of_any_surface_tolerance specified as the refined_control.

WR8: The size of the Tolerance_zone_boundary of this Composite_surface_profile_refinement_tolerance shall be less than the size of the Tolerance_zone_boundary of the Profile_of_any_surface_tolerance specified as the refined_control.

WR9: The Tolerance_zone_boundary of this Composite_surface_profile_refinement_tolerance shall be based on the same True_profile as the Tolerance_zone_boundary of the Profile_of_any_surface_tolerance specified as the refined_control.

WR10: The Composite_surface_profile_refinement_tolerance shall be specified as a refined_control by at most one Composite_surface_profile_refinement_tolerance.

NOTE EXPRESS specification:

```
*)
ENTITY composite_surface_profile_refinement_tolerance
  SUBTYPE OF (profile_of_any_surface_tolerance);
    refined_control : locating_or_refinement_surface_profile_tolerance;
END_ENTITY; -- composite_surface_profile_refinement_tolerance

( *
```

4.2.167 Composite_unit_shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Composite_unit_shape_element is a type of Composite_shape_element (see 4.2.162) that is treated as a unit, as opposed to as a group of individual Shape_element (see 4.2.810).

EXAMPLE 1 A Profile_of_any_surface_tolerance (see 4.2.747) is applied to a square cut-out that goes through a printed circuit board. The square cut-out is a Composite_unit_shape_element that consists of the four constituent Shape_elements that correspond to the four surfaces of the cut-out.

EXAMPLE 2 A Flatness_tolerance (see 4.2.386) may be applied to a Composite_unit_shape_element to specify that the coplanar surfaces that make up the Composite_unit_shape_element are required to fall within a common Tolerance_zone (see 4.2.913).

NOTE Common tolerance zones are described in 7.5 of ISO 1101.

NOTE EXPRESS specification:

```
*)  
ENTITY composite_unit_shape_element  
  SUBTYPE OF (composite_shape_element);  
END_ENTITY; -- composite_unit_shape_element  
  
(*
```

4.2.168 Concentricity_and_coaxiality_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Concentricity_and_coaxiality_tolerance is a type of Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that specifies or partially specifies the design requirement on either the concentricity or the coaxiality of the tolerated Shape_element (see 4.2.810) in relation to a Datum (see 4.2.214). Coaxiality is the quality or state of being coaxial.

NOTE Concentricity and coaxiality tolerance is defined in terms of tolerance zone definitions in 14.11 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the tolerated_shape_element shall be either a Centre_axis or a Centre_point.

NOTE The attribute tolerated_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: There shall be exactly one Datum in the Datum_system the definition of which is specified as the datum_system_definition.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

WR3: If the Shape_element specified as the tolerated_shape_element is a Centre_point, the Datum of the Datum_system the definition of which is specified as the datum_system_definition shall be a Datum_point.

WR4: If the Shape_element specified as the tolerated_shape_element is a Centre_axis, the Datum of the Datum_system the definition of which is specified as the datum_system_definition shall be a Datum_axis.

WR5: The Concentricity_and_coaxiality_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify a Diametrical_boundary_size_characteristic as its specified_boundary_size.

IP1: The form of the tolerated Shape_element shall be consistent with the form of a shape element for which the application of a concentricity and coaxiality tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY concentricity_and_coaxiality_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- concentricity_and_coaxiality_tolerance
```

(*

4.2.169 Conductance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conductance_data_element is a

The data associated with a Conductance_data_element are the following:

— additional_data.

4.2.169.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY conductance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- conductance_data_element
```

(*

4.2.170 Conductive_filled_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Conductive_filled_area` is a type of `Stratum_feature` (see 4.2.851) in which the material is conductive. A `Conductive_filled_area` may be a `Connected_filled_area` (see 4.2.184).

NOTE EXPRESS specification:

```
*)
ENTITY conductive_filled_area
  SUPERTYPE OF (connected_filled_area)
  SUBTYPE OF (stratum_feature);
END_ENTITY; -- conductive_filled_area

( *
```

4.2.171 Conductive_interconnect_element

A `Conductive_interconnect_element` is a type of `Laminate_component` (see 4.2.483). A `Conductive_interconnect_element` is either a `Conductive_interconnect_element_with_pre_defined_transitions` (see 4.2.174) or a `Conductive_interconnect_element_with_user_defined_single_transition` (see 4.2.175) or an `Unrouted_conductive_interconnect_element` (see 4.2.937). A `Conductive_interconnect_element` specifies one portion of a `Stratum_feature` (see 4.2.851) that has a sequential arrangement of `Layer_connection_point` (see 4.2.499).

NOTE A domain term for a `Conductive_interconnect_element` is a trace. The domain term may include additional aspects over that defined in this application object in certain cases. For that reason, this part of ISO 10303 does not attempt to prescribe or limit the application of the term trace by standardizing it.

EXAMPLE 1 An example of multiple `Conductive_interconnect_element`, `Land` (see 4.2.490), and `Stratum_feature_template_component` (see 4.2.862) that contribute to the definition of a `Conductor` (see 4.2.176) are shown in the partially exploded view of a `Conductor` in figure 18.

The data associated with a `Conductive_interconnect_element` are the following:

- `access_mechanisms`;
- `composed_conductor`.

4.2.171.1 access_mechanisms

The `access_mechanisms` is an inverse relationship that specifies that the existence of the `Conductive_interconnect_element` is dependent on the existence of the `Conductive_interconnect_element_terminal` (see 4.2.172) that specifies the `Conductive_interconnect_element` as its `associated_component`. There shall be at least two `Conductive_interconnect_element_terminals` for a `Conductive_interconnect_element`.

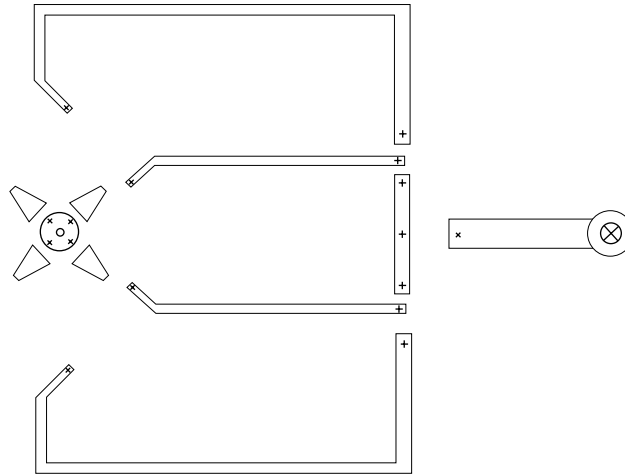


Figure 18 – Partially exploded view of a Conductor

4.2.171.2 composed_conductor

The `composed_conductor` specifies the role of the `Stratum_feature` (see 4.2.851) for the `Conductive_interconnect_element`. The `composed_conductor` need not be specified for a `Conductive_interconnect_element`.

Constraints:

WR1: The members of `Layer_connection_point` specified by a member of `Conductive_interconnect_element` shall also be specified by the feature this `interconnect_element` is composing. This is a non-exclusive rule. Other members of `Layer_connection_point` may be specified by the conductor.

NOTE EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element
  ABSTRACT SUPERTYPE OF (ONEOF (
    unrouted_conductive_interconnect_element,
    conductive_interconnect_element_with_pre_defined_transitions,
    conductive_interconnect_element_with_user_defined_single_transition))
  SUBTYPE OF (laminate_component);
  composed_conductor : OPTIONAL stratum_feature;
  INVERSE
    access_mechanisms : SET [2:?] OF conductive_interconnect_element_terminal FOR
      associated_component;
END_ENTITY; -- conductive_interconnect_element

( *
```

4.2.172 Conductive_interconnect_element_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Conductive_interconnect_element_terminal` is a type of `Component_terminal` (see 4.2.137). The `Conductive_interconnect_element_terminal` is a terminal for the trace defined by `Conductive_interconnect_element`.

NOTE This is used for detailed design and analysis, and also as a means to check the quality of geometry representation and transfer.

The data associated with a `Conductive_interconnect_element_terminal` are the following:

- `associated_component`;
- `connection`;
- `location`;
- `name`.

4.2.172.1 associated_component

The `associated_component` specifies the role of the `Conductive_interconnect_element` (see 4.2.171) for the `Conductive_interconnect_element_terminal`.

4.2.172.2 connection

The `connection` specifies the role of the `Connection_zone` (see 4.2.185) for the `Conductive_interconnect_element_terminal`. The `connection` is optional and need not be supplied if the data is not available in the source system. If the data is available in the source system, it shall be supplied.

4.2.172.3 location

The `location` is an inverse attribute that specifies the `Design_view_terminal_component_shape_relationship` that the `Conductive_interconnect_element_terminal` is existence dependent on. There may be one or two `Design_view_terminal_component_shape_relationship` depending on whether there is a single shape representation for the design or whether there is a two dimensional shape and a three dimensional shape.

4.2.172.4 name

The `name` specifies the string for the human-interpretable identifier for the `Conductive_interconnect_element_terminal`.

Constraints:

UR1: The combination of name and associated_component inherited from Component_feature shall be unique within a population of Conductive_interconnect_element_terminal.

WR1: The same Stratum shall be referenced by the path starting at location and the path starting at associated_component inherited from the Component_feature.

NOTE EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element_terminal
  SUBTYPE OF (laminate_component_feature);
    SELF\component_feature.associated_component : conductive_interconnect_element;
    connection                                : OPTIONAL connection_zone;
    name                                       : STRING;
    location : layer_connection_point;
UNIQUE
  UR1: name, SELF\component_feature.associated_component;
WHERE
  WR1: NOT(EXISTS(SELF\component_feature.
    associated_component\conductive_interconnect_element.
    composed_conductor)) OR
    (location.resident_design_layer_stratum = SELF\component_feature.
    associated_component\conductive_interconnect_element.
    composed_conductor.resident_stratum);
END_ENTITY; -- conductive_interconnect_element_terminal
(

```

4.2.173 Conductive_interconnect_element_terminal_link

A Conductive_interconnect_element_terminal_link is a type of Managed_design_object (see 4.2.524). A Conductive_interconnect_element_terminal_link is a directed link between two Conductive_interconnect_element_terminal (see 4.2.172) in order to support the sequence of Conductive_interconnect_element_terminal required in Conductive_interconnect_element_with_pre_defined_transitions (see 4.2.174).

The data associated with a Conductive_interconnect_element_terminal_link are the following:

- associated_element;
- precedent_point;
- subsequent_point.

4.2.173.1 associated_element

An associated_element specifies the role of the Conductive_interconnect_element_with_pre_defined_transitions (see 4.2.174) for a Conductive_interconnect_element_terminal_link.

4.2.173.2 precedent_point

A precedent_point specifies one role of the Conductive_interconnect_element_terminal (see 4.2.172) for a Conductive_interconnect_element_terminal_link.

4.2.173.3 subsequent_point

A subsequent_point specifies one role of the Conductive_interconnect_element_terminal (see 4.2.172) for a Conductive_interconnect_element_terminal_link.

Constraints:

WR1: The precedent_point shall not be the subsequent_point.

WR2: The precedent_point shall be on the associated_element.

WR3: The subsequent_point shall be on the associated_element.

NOTE EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element_terminal_link
  SUBTYPE OF (managed_design_object);
    associated_element : conductive_interconnect_element_with_pre_defined_transitions;
    precedent_point    : conductive_interconnect_element_terminal;
    subsequent_point   : conductive_interconnect_element_terminal;
WHERE
  WR1 : precedent_point &#60;&#62; subsequent_point;
  WR2 : precedent_point.associated_component = associated_element;
  WR3 : subsequent_point.associated_component = associated_element;
END_ENTITY; -- conductive_interconnect_element_terminal_link
( *
```

4.2.174 Conductive_interconnect_element_with_pre_defined_transitions

A `Conductive_interconnect_element_with_pre_defined_transitions` is a type of `Conductive_interconnect_element` (see 4.2.171). The data provided by `Conductive_interconnect_element_with_pre_defined_transitions` is typical data available from two dimensional CAD applications. The data does not include explicit representation of the variability in the vertical dimension that occurs in a real "planar" product due to material build up of successive layers of material. The `usage_definition` is required to be orthogonal to the `centreline_shape_representation`, but this is insufficient information in a three-dimensional layout. In a two-dimensional layout, the `usage_definition` shape x direction vector is constrained to be parallel with the x direction vector of the shape of the `Stratum_feature` (see 4.2.851) referenced by the inherited attribute `composed_conductor`. In a two-dimensional design, all `Stratum_features` are parallel. In a three-dimensional design, the same parallelism constraint applies, but the underlying substrate is not presumed to be planar. The underlying surface shape idealization is contained in the `Stratum_surface_shape` (see 4.2.870) or `Stratum_average_surface_shape` (see 4.2.848). The `Stratum_feature_shape_stratum_surface_shape_relationship` (see 4.2.859) or `Stratum_feature_shape_stratum_average_surface_shape_relationship` (see 4.2.858) shall be populated to specify to a processor the design intent. The `centreline_shape_representation` shall be a subset of either the points in the `Stratum_average_surface_shape` or of the points in the `Stratum_surface_shape`.

NOTE The "pre-defined transition" refers to the extrusion of the `trace_template` shape (a cross-sectional definition) along the path defined by its `centreline_shape_representation` (a `Curve` (see 4.2.203)).

The data associated with a `Conductive_interconnect_element_with_pre_defined_transitions` are the following:

- `centreline_shape_representation`;
- `connected_point`;
- `usage_definition`.

4.2.174.1 centreline_shape_representation

The `centreline_shape_representation` specifies the `Curve` representing the centreline of the `Conductive_interconnect_element_with_pre_defined_transitions`. The centreline when combined with the `usage_definition` provides a CAD abstraction of the physical shape.

4.2.174.2 connected_point

The `connected_points` is an inverse relationship that specifies that the `Conductive_interconnect_element_with_pre_defined_transitions` is existence dependent on at least one `Conductive_interconnect_element_terminal_link` (see 4.2.173) for the `Conductive_interconnect_element`.

4.2.174.3 usage_definition

The usage_definition specifies the role of the Trace_template (see 4.2.930) for the Conductive_interconnect_element_with_pre_defined_transitions. The usage_definition is a cross section definition. When combined with the centreline shape, a complete CAD abstraction is formed for the two-dimensional case. The three-dimensional case requires additional data representing the abstract surface in which the conductor being defined will be located.

NOTE The Trace_template describes the cross-section parameters required by a processor that adds material along the path described by the centreline.

NOTE EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element_with_pre_defined_transitions
  SUPERTYPE OF (join_two_physical_connectivity_definition_supporting_printed_component)
  SUBTYPE OF (conductive_interconnect_element);
  SELF\assembly_component.usage_definition : trace_template;
  centreline_shape                          : curve;
  SELF\conductive_interconnect_element.
    composed_conductor                      : stratum_feature;
INVERSE
  connected_points                          : SET [1:?] OF conductive_interconnect_element
                                           for associated_element;
WHERE
  WR1: SIZEOF(connected_points) =
        SIZEOF(SELF\conductive_interconnect_element.access_mechanisms) -1;
  WR2: correlated_terminals(SELF\conductive_interconnect_element.
    access_mechanisms,connected_points);
END_ENTITY; -- conductive_interconnect_element_with_pre_defined_transitions
(

```

4.2.175 Conductive_interconnect_element_with_user_defined_single_transition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conductive_interconnect_element_with_user_defined_single_transition is a type of Conductive_interconnect_element (see 4.2.171). The Conductive_interconnect_element_with_user_defined_single_transition supports the specification of custom start and end shape templates and separately specified left_curve_shape_representation and right_curve_shape_representation (which are either Bounding_curve or Trimmed_line).

NOTE 1 For examples of the curves that can be used, see Bounding_curve (see 4.2.80).

NOTE 2 The electrical properties of the Conductive_interconnect_element_with_user_defined_single_transition may be assigned using a requirement allocation structure.

EXAMPLE 1 Figure 12 illustrates a detail of a layout drawing that shows a transition on a trace. The segment "I" is supported in this part of ISO 10303 as a member of `Conductive_interconnect_element_with_user_defined_single_transition`.

The data associated with a `Conductive_interconnect_element_with_user_defined_single_transition` are the following:

- `end_definition`;
- `end_terminus` ;
- `left_curve_shape_representation` ;
- `right_curve_shape_representation` ;
- `start_definition` ;
- `start_terminus` .

4.2.175.1 `end_definition`

The `end_definition` specifies one role of the `Trace_template` (see 4.2.930) for the `Conductive_interconnect_element_with_user_defined_single_transition`.

4.2.175.2 `end_terminus`

The `end_terminus` specifies one role of the `Layer_connection_point` (see 4.2.499) for the `Conductive_interconnect_element_with_user_defined_single_transition`.

4.2.175.3 `left_curve_shape_representation`

The `left_curve_shape_representation` specifies one role of either a `Bounding_curve` (see 4.2.80) or `Trimmed_line` (see 4.2.933) for the `Conductive_interconnect_element_with_user_defined_single_transition`.

4.2.175.4 `right_curve_shape_representation`

The `right_curve_shape_representation` specifies one role of either a `Bounding_curve` (see 4.2.80) or `Trimmed_line` (see 4.2.933) for the `Conductive_interconnect_element_with_user_defined_single_transition`.

4.2.175.5 `start_definition`

The `start_definition` specifies one role of the `Trace_template` (see 4.2.930) for the `Conductive_interconnect_element_with_user_defined_single_transition`.

4.2.175.6 start_terminus

The start_terminus specifies one role of the Layer_connection_point (see 4.2.499) for the Conductive_interconnect_element_with_user_defined_single_transition.

NOTE EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element_with_user_defined_single_transition
  SUBTYPE OF (conductive_interconnect_element);
  SELF\conductive_interconnect_element.
    composed_conductor      : stratum_feature;
    start_definition        : trace_template;
    end_definition          : trace_template;
    left_curve_shape_representation : trimmed_or_bounding_curve;
    right_curve_shape_representation : trimmed_or_bounding_curve;
    start_terminus          : layer_connection_point;
    end_terminus            : layer_connection_point;
END_ENTITY; -- conductive_interconnect_element_with_user_defined_single_transition

( *
```

4.2.176 Conductor

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conductor is a type of Stratum_feature (see 4.2.851) realized in a Design_layer_stratum (see 4.2.275) that participates in the implementation of a Physical_network (see 4.2.660).

EXAMPLE 1 Figure 19 illustrates a detail of a layout drawing that shows the Conductor in relationship to a component at the next level of the assembly.

The data associated with a Conductor are the following:

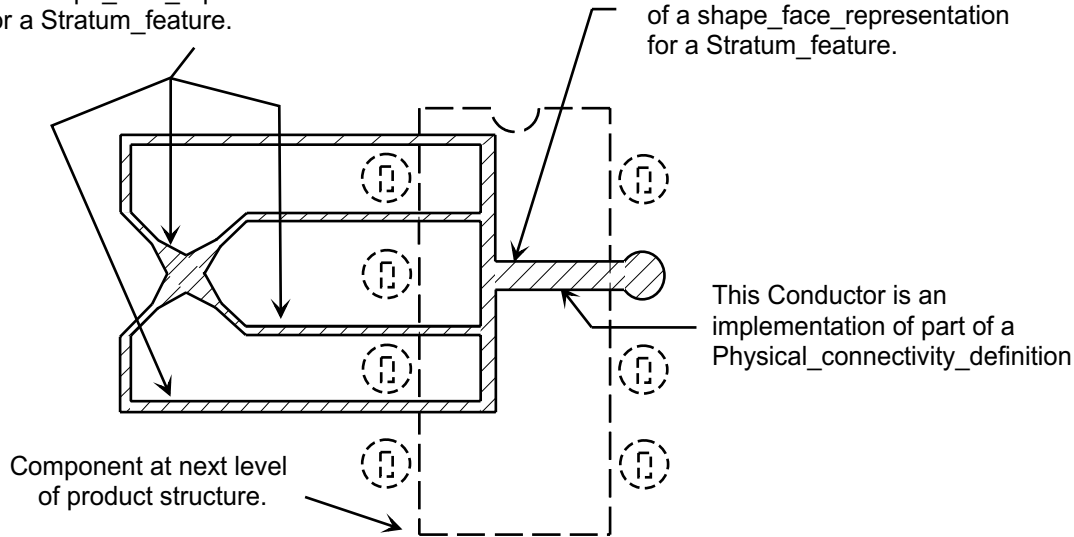
— implementation_requirement.

4.2.176.1 implementation_requirement

The implementation_requirement specifies the Intra_stratum_join_relationship (see 4.2.477) that is used to constrain the Conductor by specifying the Layer_connection_point (see 4.2.499) that are to be connected by the Conductor.

A Feature_shape_definition that is a Curve that is a Closed_curve in the role of face_inner_bound for a Planar_bound_face. This is in the role of a shape_face_representation for a Stratum_feature.

A Feature_shape_definition that is a Curve that is a Closed_curve in the role of face_outer_bound for a Planar_bound_face. This is in the role of a shape_face_representation for a Stratum_feature.



Four traces (domain term) converging on a location separately, in order to avoid three obstructions. All participate in the same implementation of a connectivity requirement.

Figure 19 – Conductor

Constraints:

WR1: The shape that is a Stratum_feature shape shall not violate the nearest approach constraint to another shape assigned to the same design layer and realized as the stratum material of the design layer stratum.

WR2: A Conductor shall enclose two or more layer connection point in the region between or on the outer bound and an inner bound; or inside or on the outer bound that has no inner bound.

NOTE EXPRESS specification:

```

*)
ENTITY conductor
  SUBTYPE OF (stratum_feature);
--tt tc1 wd9 added
  SELF\stratum_feature.resident_stratum      : design_layer_stratum;
  implementation_requirement : intra_stratum_join_relationship;
END_ENTITY; -- conductor

( *

```

4.2.177 Configuration_controlled_message

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Configuration_controlled_message is a type of Managed_design_object (see 4.2.524). A Configuration_controlled_message is a message that has its content under version control, and that is source data for application of marking materials to a product.

The data associated with a Configuration_controlled_message are the following:

- message_content;
- message_source;
- message_type_specification.

4.2.177.1 message_content

The message_content specifies the text in the Configuration_controlled_message.

4.2.177.2 message_source

The message_source specifies either an Ee_approval (see 4.2.314), an Ee_product_version (see 4.2.328), or an Organization (see 4.2.569) for the Configuration_controlled_message.

4.2.177.3 message_type_specification

The message_type_specification specifies either an approval_date, a manufacturer_code, a part_number, or a revision_identifier for the Configuration_controlled_message. A message_type_specification is necessary to differentiate among the possible message sources to identify the one that has been selected for the literal string.

Constraints:

WR1: If message_type_specification is "approval_date" then message_content is equal to Ee_approval.signoff_date.

WR2: If message_type_specification is "manufacturer_code" then message_content is equal to Organization.identifier.

WR3: If message_type_specification is "part_number" then message_content is equal to Ee_product.part_number.

WR4: If message_type_specification is "revision_code" then message_content is equal to Ee_product_version.revision_code.

NOTE EXPRESS specification:

```

*)
ENTITY configuration_controlled_message
  SUBTYPE OF (managed_design_object);
    message_source      : message_source_select;
    message_content     : STRING;
    message_type_specification : message_type;
END_ENTITY; -- configuration_controlled_message

( *
```

4.2.178 Conical_tolerance_zone_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conical_tolerance_zone_boundary is a type of Tolerance_zone_boundary (see 4.2.915) that is conical in form. The shape of the Conical_tolerance_zone_boundary is defined by a frustum the size of which is specified at the surfaces that define the ends of the frustum.

The data associated with a Conical_tolerance_zone_boundary are the following:

— boundary_and_surface_relationships.

4.2.178.1 boundary_and_surface_relationships

The boundary_and_surface_relationships is an inverse relationship that specifies that the existence of the Conical_tolerance_zone_boundary is dependent on the existence of two Conical_tolerance_zone_boundary_and_surface_relationship (see 4.2.179) that specify the Conical_tolerance_zone_boundary as their sized_tolerance_zone_boundary. There shall be exactly two Conical_tolerance_zone_boundary_and_surface_relationship for a Conical_tolerance_zone_boundary.

NOTE EXPRESS specification:

```

*)
ENTITY conical_tolerance_zone_boundary
  SUBTYPE OF (tolerance_zone_boundary);
  INVERSE
    boundary_and_surface_relationships : SET [2:2] OF
    conical_tolerance_zone_boundary_and_surface_relationship FOR
    sized_tolerance_zone_boundary;
END_ENTITY; -- conical_tolerance_zone_boundary

( *
```

4.2.179 Conical_tolerance_zone_boundary_and_surface_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conical_tolerance_zone_boundary_and_surface_relationship is a type of Managed_design_object (see 4.2.524). A Conical_tolerance_zone_boundary_and_surface_relationship is a relationship that associates a Conical_tolerance_zone_boundary (see 4.2.178) with one of the two Physical_feature_or_part_template (see 4.2.658) at which the Conical_tolerance_zone_boundary diameter is specified.

The data associated with a Conical_tolerance_zone_boundary_and_surface_relationship are the following:

- size_characteristic;
- sized_tolerance_zone_boundary;
- terminating_surface.

4.2.179.1 size_characteristic

The size_characteristic is an inverse relationship that specifies that the existence of the Conical_tolerance_zone_boundary_and_surface_relationship is dependent on the existence of the Conical_tolerance_zone_boundary_diametrical_size_characteristic (see 4.2.181) that specifies the Conical_tolerance_zone_boundary_and_surface_relationship as its sized_boundary_at_surface.

4.2.179.2 sized_tolerance_zone_boundary

The sized_tolerance_zone_boundary specifies the Conical_tolerance_zone_boundary (see 4.2.178) the diameter of which is specified at the Physical_feature_or_part_template (see 4.2.658) specified as the terminating_surface.

4.2.179.3 terminating_surface

The terminating_surface specifies the Physical_feature_or_part_template (see 4.2.658) at which the diameter is specified for the Conical_tolerance_zone_boundary (see 4.2.178).

NOTE EXPRESS specification:

```

*)
ENTITY conical_tolerance_zone_boundary_and_surface_relationship
  SUBTYPE OF (managed_design_object);
    sized_tolerance_zone_boundary : conical_tolerance_zone_boundary;
    terminating_surface           : physical_feature_or_part_template;
  INVERSE
    size_characteristic :
conical_tolerance_zone_boundary_diametrical_size_characteristic FOR
sized_boundary_at_surface;
END_ENTITY; -- conical_tolerance_zone_boundary_and_surface_relationship

( *
```

4.2.180 Conical_tolerance_zone_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conical_tolerance_zone_boundary_definition is a type of Tolerance_zone_boundary_definition (see 4.2.916) that characterizes a Conical_tolerance_zone_boundary (see 4.2.178).

The data associated with a Conical_tolerance_zone_boundary_definition are the following:

- boundary_sizes;
- defined_zone_boundary.

4.2.180.1 boundary_sizes

The boundary_sizes specifies the Conical_tolerance_zone_boundary_diametrical_size_characteristic (see 4.2.181) that defines the size of the Conical_tolerance_zone_boundary (see 4.2.178) at specified Physical_feature_or_part_templates. There shall be exactly two Conical_tolerance_zone_boundary_diametrical_size_characteristic specified as the boundary_sizes for the Conical_tolerance_zone_boundary_definition.

4.2.180.2 defined_zone_boundary

The defined_zone_boundary specifies the Conical_tolerance_zone_boundary (see 4.2.178) that is characterized by the Conical_tolerance_zone_boundary_definition.

Constraints:

WR1: Each Geometric_tolerance that defines a Conical_tolerance_zone_boundary_diametrical_size_-characteristic of the Conical_tolerance_zone_boundary_definition shall be of the same type.

WR2: Each Geometric_tolerance that defines a Conical_tolerance_zone_boundary_diametrical_size_-characteristic of the Conical_tolerance_zone_boundary_definition shall reference the same Datum_system_defining_conditions.

WR3: Each Geometric_tolerance that defines a Conical_tolerance_zone_boundary_diametrical_size_-characteristic of the Conical_tolerance_zone_boundary_definition shall specify the same value for the applied_material_condition_property_name.

WR4: Each Geometric_tolerance that defines a Conical_tolerance_zone_boundary_diametrical_size_-characteristic of the Conical_tolerance_zone_boundary_definition shall specify the same instance of Centre_axis as the toleranced_shape_element.

WR5: The Conical_tolerance_zone_boundary_definition shall be specified as the boundary_definition by a Conical_tolerance_zone_definition.

NOTE EXPRESS specification:

```

*)
ENTITY conical_tolerance_zone_boundary_definition
  SUBTYPE OF (tolerance_zone_boundary_definition);
    boundary_sizes :
SET [2:2] OF conical_tolerance_zone_boundary_diametrical_size_characteristic;
  SELF\tolerance_zone_boundary_definition.defined_zone_boundary :
conical_tolerance_zone_boundary;
END_ENTITY; -- conical_tolerance_zone_boundary_definition

```

(*

4.2.181 Conical_tolerance_zone_boundary_diametrical_size_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conical_tolerance_zone_boundary_diametrical_size_characteristic is a type of Managed_design_object (see 4.2.524). A Conical_tolerance_zone_boundary_diametrical_size_characteristic is a definition that characterizes the size of a Conical_tolerance_zone_boundary (see 4.2.178) at one of the Physical_feature_or_part_template (see 4.2.658) that defines the ends of the Conical_tolerance_zone_boundary.

The data associated with a `Conical_tolerance_zone_boundary_diametrical_size_characteristic` are the following:

- `boundary_definition`;
- `sized_boundary_at_surface`;
- `tolerance_value`.

4.2.181.1 `boundary_definition`

The `boundary_definition` is an inverse relationship that specifies that the existence of the `Conical_tolerance_zone_boundary_diametrical_size_characteristic` is dependent on the existence of the `Conical_tolerance_zone_boundary_definition` (see 4.2.180) that specifies the `Conical_tolerance_zone_boundary_diametrical_size_characteristic` as one of its `boundary_sizes`.

4.2.181.2 `sized_boundary_at_surface`

The `sized_boundary_at_surface` specifies the `Conical_tolerance_zone_boundary_and_surface_relationship` (see 4.2.179) that associates the `Conical_tolerance_zone_boundary` (see 4.2.178) with a `Physical_feature_or_part_template` (see 4.2.658) that defines an end of the `Conical_tolerance_zone_boundary` at which the size is characterized by the `Conical_tolerance_zone_boundary_diametrical_size_characteristic`.

4.2.181.3 `tolerance_value`

The `tolerance_value` specifies the `Length_data_element` (see 4.2.511) that represents the diameter of the specified `Conical_tolerance_zone_boundary` (see 4.2.178) at the specified `Physical_feature_or_part_template` (see 4.2.658) for the `Conical_tolerance_zone_boundary_diametrical_size_characteristic`.

Constraints:

WR1: The `Conical_tolerance_zone_boundary` specified as the `sized_tolerance_zone_boundary` by the `Conical_tolerance_zone_boundary_and_surface_relationship` that is specified as the `sized_boundary_at_surface` shall be the same instance of `Conical_tolerance_zone_boundary` as specified as the `defined_zone_boundary` by the `Conical_tolerance_zone_boundary_definition` that is specified as the `boundary_definition`. That is, the `Conical_tolerance_zone_boundary_diametrical_size_characteristic` and the `Conical_tolerance_zone_boundary_definition` specified as the `boundary_definition` shall define the same `Conical_tolerance_zone_boundary`.

WR2: The `Conical_tolerance_zone_boundary_diametrical_size_characteristic` shall be specified by exactly one `Geometric_tolerance` as the `tolerance_zone_or_boundary_definition`, and that `Geometric_tolerance` shall be one of a `Position_tolerance`, a `Parallelism_tolerance`, a `Perpendicularity_tolerance`, or an `Angularity_tolerance`.

NOTE EXPRESS specification:

```

*)
ENTITY conical_tolerance_zone_boundary_diametrical_size_characteristic
  SUBTYPE OF (managed_design_object);
    sized_boundary_at_surface : conical_tolerance_zone_boundary_and_surface_relationship;
    tolerance_value           : length_data_element;
  INVERSE
    boundary_definition : conical_tolerance_zone_boundary_definition FOR
                          boundary_sizes;
END_ENTITY; -- conical_tolerance_zone_boundary_diametrical_size_characteristic

( *
```

4.2.182 Conical_tolerance_zone_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Conical_tolerance_zone_definition is a type of Tolerance_zone_definition (see 4.2.920) the boundary of which is defined by a Conical_tolerance_zone_boundary_definition (see 4.2.180). A Conical_tolerance_zone_definition is defined when it is desired to place a more restrictive specification on one end of a Shape_element (see 4.2.810) than on the other end.

NOTE Conical tolerance zones are described in 5.8 of ASME Y14.5M.

The data associated with a Conical_tolerance_zone_definition are the following:

— boundary_definition.

4.2.182.1 boundary_definition

The boundary_definition specifies the Conical_tolerance_zone_boundary_definition (see 4.2.180) that defines the boundary for the Conical_tolerance_zone_definition.

Constraints:

WR1: The Conical_tolerance_zone_definition shall not be specified as the tolerance_zone_or_boundary_definition by a Geometric_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY conical_tolerance_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
  SELF\tolerance_zone_definition.boundary_definition :
conical_tolerance_zone_boundary_definition;
END_ENTITY; -- conical_tolerance_zone_definition

--tt tc1 wd9
( *

```

4.2.183 Connected_area_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connected_area_component is a type of Stratum_feature_template_component (see 4.2.862).

NOTE EXPRESS specification:

```

*)
ENTITY connected_area_component
  SUBTYPE OF (stratum_feature_template_component);
--tt tc1 wd9 deleted following
-- SELF\assembly_component.usage_definition : stratum_feature_template;
-- SELF\stratum_feature_template_component.stratum_feature_implementation : stratum_fea
associated_layer_connection_point : SET [1:?] OF layer_connection_point;
WHERE
  WR1: NOT(EXISTS(SELF\assembly_component.usage_definition));
END_ENTITY; -- connected_area_component

( *

```

4.2.184 Connected_filled_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connected_filled_area is a type of Conductive_filled_area (see 4.2.170). The Connected_filled_area is a Conductive_filled_area (see 4.2.170) with which a set of layer connection points are associated (as specified in the implementation_requirement attribute).

The data associated with a Connected_filled_area are the following:

— implementation_requirement.

4.2.184.1 implementation_requirement

The implementation_requirement specifies the Intra_stratum_join_relationship (see 4.2.477) for which the Connected_filled_area is generated.

NOTE EXPRESS specification:

```
*)
ENTITY connected_filled_area
  SUBTYPE OF (conductive_filled_area);
  --tt tc1 wd9 added
    SELF\stratum_feature.resident_stratum      : design_layer_stratum;
    implementation_requirement : intra_stratum_join_relationship;
END_ENTITY; -- connected_filled_area

( *
```

4.2.185 Connection_zone

A Connection_zone is a type of Managed_design_object (see 4.2.524). A Connection_zone is an area on the surface of a design object that may be used to establish electrical or physical connectivity with another design object. A Connection_zone shape is defined in the shape definition of the feature that references the Connection_zone. Since more than one feature may reference the same shape but for different positions, more than one feature may reference the same Connection_zone.

The data associated with a Connection_zone are the following:

- zone_shape_1;
- zone_shape_2;
- representation_colour.

4.2.185.1 zone_shape_1

The zone_shape_1 is an inverse relationship that specifies that the Connection_zone is existent dependent on the Connection_zone_shape (see ??) that specifies the Connection_zone. There shall be zero or more zone_shape_1 for a Connection_zone.

4.2.185.2 zone_shape_2

The zone_shape_2 is an inverse relationship that specifies that the Connection_zone is existent dependent on the Connection_zone_shape (see ??) that specifies the Connection_zone. There shall be zero or more zone_shape_2 for a Connection_zone.

4.2.185.3 representation_colour

The representation_colour specifies the Ee_colour (see 4.2.315) that depicts the area constituting the Connection_zone. The representation_colour need not be specified.

Constraints:

WR1: All points in a Connection_zone shall be topologically connected.

WR2: A particular instance of Connection_zone shall be referenced by at least one shape instantiation.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone
  SUBTYPE OF (managed_design_object);
  representation_colour : OPTIONAL ee_colour;
INVERSE
  zone_shape_1          : SET [0:1] OF
    design_view_connection_zone_position_relationship for
    associated_usage;
  zone_shape_2          : SET [0:1] OF
    usage_view_connection_zone_terminal_shape_relationship for
    associated_usage;
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'AP210_ARM.ASSEMBLY_MODULE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.BARE_DIE_TERMINAL.EXTERNAL_CONNECTION_AREA')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.CONDUCTIVE_INTERCONNECT_ELEMENT_TERMINAL.CONNECTION')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.INTERCONNECT_COMPONENT_INTERFACE_TERMINAL.CONNECTION')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.INTERCONNECT_COMPONENT_JOIN_TERMINAL.CONNECTION')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.INTERCONNECT_MODULE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.LAND_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.PACKAGE_TERMINAL.EXTERNAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.PACKAGE_TERMINAL.INTERNAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.PRINTED_PART_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.VIA_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) +
    SIZEOF(USEDIN(SELF, 'AP210_ARM.PART_MOUNTING_FEATURE.MOUNTING_AREA')) &#62;= 0;
  WR2: SIZEOF(zone_shape_1) + SIZEOF(zone_shape_2) = 1;
END_ENTITY; -- connection_zone

( *
```

4.2.186 Connection_zone_based_assembly_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Connection_zone_based_assembly_joint` is a type of `Assembly_joint` (see 4.2.41). A `Connection_zone_based_assembly_joint` consists of a `zone_1` and a `zone_2`.

The data associated with a `Connection_zone_based_assembly_joint` are the following:

- `zone_1`;
- `zone_2`.

4.2.186.1 `zone_1`

A `zone_1` specifies one role of a `Connection_zone` (see 4.2.185) for a `Connection_zone_based_assembly_joint`.

4.2.186.2 `zone_2`

A `zone_2` specifies one role of a `Connection_zone` (see 4.2.185) for a `Connection_zone_based_assembly_joint`.

NOTE There are no semantics associated with the sequence of zones in a `Connection_zone_based_assembly_joint`.

Constraints:

WR1: The `zone_1` shall not be `zone_2`.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone_based_assembly_joint
  SUBTYPE OF (assembly_joint);
    zone_1 : connection_zone;
    zone_2 : connection_zone;
WHERE
  WR1: zone_1 :&#60;&#62;: zone_2;
(* The zone_1 shall be referenced by assembled_features_1 inherited from
the assembly_joint supertype either directly or indirectly.
The zone_1 shall not be referenced by assembled_features_2 inherited from
the assembly_joint supertype either directly or indirectly.
The zone_2 shall be referenced by assembled_features_2 inherited from
the assembly_joint supertype either directly or indirectly.
The zone_2 shall not be referenced by assembled_features_1 inherited from
the assembly_joint supertype either directly or indirectly. *)
END_ENTITY; -- connection_zone_based_assembly_joint

( *
```

4.2.187 Connection_zone_based_fabrication_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connection_zone_based_fabrication_joint is a type of Fabrication_joint (see 4.2.368). A Connection_zone_based_fabrication_joint consists of a zone_1 and a zone_2.

The data associated with a Connection_zone_based_fabrication_joint are the following:

- zone_1;
- zone_2.

4.2.187.1 zone_1

A zone_1 specifies one role of a Connection_zone (see 4.2.185) for a Connection_zone_based_fabrication_joint.

4.2.187.2 zone_2

A zone_2 specifies one role of a Connection_zone (see 4.2.185) for a Connection_zone_based_fabrication_joint.

NOTE There are no semantics associated with the sequence of zones in a Connection_zone_based_fabrication_joint.

Constraints:

WR1: The zone_1 shall not be zone_2.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone_based_fabrication_joint
  SUBTYPE OF (fabrication_joint);
    zone_1 : connection_zone;
    zone_2 : connection_zone;
WHERE
  WR1: zone_1 :&#60;&#62;: zone_2;
(* The zone_1 shall be referenced by fabricated_features_1 inherited from
the fabrication_joint supertype either directly or indirectly.
The zone_1 shall not be referenced by fabricated_features_2 inherited from
the fabrication_joint supertype either directly or indirectly.
The zone_2 shall be referenced by fabricated_features_2 inherited from
the fabrication_joint supertype either directly or indirectly.
The zone_2 shall not be referenced by fabricated_features_1 inherited from
the fabrication_joint supertype either directly or indirectly. *)
END_ENTITY; -- connection_zone_based_fabrication_joint

( *
```

4.2.188 Connection_zone_interface_plane_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connection_zone_interface_plane_relationship is a type of Managed_design_object (see 4.2.524). A Connection_zone_interface_plane_relationship establishes a product structure relationship that may be represented in a geometric context for the purpose of deriving a 2D or 3D representation of the intersection of a terminal interface.

The data associated with a Connection_zone_interface_plane_relationship are the following:

- interface_plane;
- mating_zone;
- name.

4.2.188.1 interface_plane

An interface_plane specifies the role of a Seating_plane (see 4.2.802) for the Connection_zone_interface_plane_relationship.

4.2.188.2 mating_zone

A mating_zone specifies the role of a Connection_zone (see 4.2.185) for the Connection_zone_interface_plane_relationship.

4.2.188.3 name

A name specifies the string that identifies the Connection_zone_interface_plane_relationship.

Constraints:

UR1: The name shall be unique within a population of Connection_zone_interface_plane_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone_interface_plane_relationship
  SUBTYPE OF (managed_design_object);
    interface_plane      : seating_plane;
    mating_zone          : connection_zone;
    derived_zone         : connection_zone;
    derived_zone_shape_class : STRING;
INVERSE
    associated_package_terminal : SET [0:1] OF package_terminal FOR
                                   seating_plane_zone;
    associated_bare_die_terminal : SET [0:1] OF bare_die_terminal FOR
                                   seating_plane_zone;
UNIQUE
    UR1: interface_plane, mating_zone;
WHERE
    WR1: derived_zone_shape_class IN ['area', 'edge'];
    WR2: NOT((SIZEOF(associated_package_terminal) = 1) AND
              (SIZEOF(associated_bare_die_terminal) = 1));
END_ENTITY; -- connection_zone_interface_plane_relationship
(

```

4.2.189 Connection_zone_map_identification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connection_zone_map_identification is a type of Managed_design_object (see 4.2.524). A Connection_zone_map_identification specifies the geometric, causal, relationship between a reference Connection_zone associated with a Package_terminal and a mapped Connection_zone associated with a template interface terminal.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone_map_identification
  SUBTYPE OF (managed_design_object);
    reference_connection_zone_shape : usage_view_connection_zone_terminal_shape_relationship;
    mapped_connection_zone_shape : usage_view_connection_zone_terminal_shape_relationship;
UNIQUE
  UR1: reference_connection_zone_shape, mapped_connection_zone_shape;
WHERE
  WR1: reference_connection_zone_shape &#60;&#62;
    mapped_connection_zone_shape;
  WR2: reference_connection_zone_shape.associated_connection_zone_shape_definition &#60;&#62;
    mapped_connection_zone_shape.associated_connection_zone_shape_definition;
END_ENTITY; -- connection_zone_map_identification

( *
```

4.2.190 Connection_zone_model_port_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connection_zone_model_port_assignment is a type of Managed_design_object (see 4.2.524). A Connection_zone_model_port_assignment is the relationship between the Analytical_model_port (see 4.2.17) and the Connection_zone (see 4.2.185), for the purpose of mapping the connection points to the model ports directly.

NOTE 1 In some circumstances it may be necessary to assign the Connection_zones to model ports in order to characterize the functionality of the terminals more accurately.

NOTE 2 Part_device_terminal_model_port_assignment (see 4.2.608) may be used in conjunction with Part_feature_model_port_assignment (see 4.2.611) where this level of detail is not required.

The data associated with a Connection_zone_model_port_assignment are the following:

- source;
- target.

4.2.190.1 source

The source specifies the role of the Connection_zone (see 4.2.185) for the Connection_zone_model_port_assignment.

4.2.190.2 target

The target specifies the role of the Analytical_model_port (see 4.2.17) for the Connection_zone_model_port_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY connection_zone_model_port_assignment
  SUBTYPE OF (managed_design_object);
    source : connection_zone;
    target : analytical_model_port;
END_ENTITY; -- connection_zone_model_port_assignment

( *

```

4.2.191 Connectivity_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connectivity_allocation is a type of Managed_design_object (see 4.2.524). A Connectivity_allocation is the relationship between a functional connectivity requirement and its implementation. The collection of functional terminals in the functional decomposition that are connected is allocated to one equivalent physical node. That equivalent physical node is an implementation in the Connectivity_allocation.

NOTE This part of ISO 10303 supports the capability to specify the paths through the functional decomposition that are reachable via terminals associated with given nodes. This is a statement of transitive closure for connectivity across the decomposition. This part of ISO 10303 does not separately represent a fully resolved functional node.

The data associated with a Connectivity_allocation are the following:

- connectivity_requirement;
- implementation.

4.2.191.1 connectivity_requirement

The connectivity_requirement specifies the role of the Aggregate_connectivity_requirement (see 4.2.7) for the Connectivity_allocation.

NOTE The connectivity_requirement effectively specifies one equivalent functional node.

4.2.191.2 implementation

The implementation specifies the role of the Physical_connectivity_definition (see 4.2.652) for the Connectivity_allocation.

Constraints:

UR1: The implementation shall be unique within a population of Connectivity_allocations.

NOTE EXPRESS specification:

```

*)
ENTITY connectivity_allocation
  SUBTYPE OF (managed_design_object);
    implementation      : physical_connectivity_definition;
    connectivity_requirement : aggregate_connectivity_requirement;
  UNIQUE
    UR1 : implementation;
END_ENTITY; -- connectivity_allocation

--tt tc1 wd12 added
( *

```

4.2.192 Connectivity_allocation_to_physical_network

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Connectivity_allocation_to_physical_network is a

The data associated with a Connectivity_allocation_to_physical_network are the following:

— additional_data.

4.2.192.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY connectivity_allocation_to_physical_network
  SUBTYPE OF (managed_design_object);
    implementation      : physical_network;
    connectivity_requirement : aggregate_connectivity_requirement;
  UNIQUE
    UR1 : implementation;
END_ENTITY; -- connectivity_allocation

( *

```


4.2.193 Constraint_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Constraint_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that limits possible solutions to the requirements specification by identifying boundary conditions for any potential solution. A Constraint_occurrence may be one of Ee_product_definition_based_constraint_occurrence (see 4.2.324) or of Rule_based_constraint_occurrence (see 4.2.782) or may be a combination of the two.

NOTE EXPRESS specification:

```
* )
ENTITY constraint_occurrence
  ABSTRACT SUPERTYPE OF
    (ee_product_definition_based_constraint_occurrence
     ANDOR rule_based_constraint_occurrence)
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- constraint_occurrence
( *
```

4.2.194 Contact_size_dependent_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Contact_size_dependent_land is a type of Land (see 4.2.490) whose shape is dependent on the size of the contact area of the Component_terminal (see 4.2.137) that will be joined to the Contact_size_dependent_land in the assembly.

EXAMPLE 1 Figure 20 is a detail that illustrates the Contact_size_dependent_land in relationship to the terminal that mates with it.

The data associated with a Contact_size_dependent_land are the following:

— usage_definition.

4.2.194.1 usage_definition

The usage_definition specifies the role of the Default_attachment_size_based_land_physical_template (see 4.2.247) for the Contact_size_dependent_land.

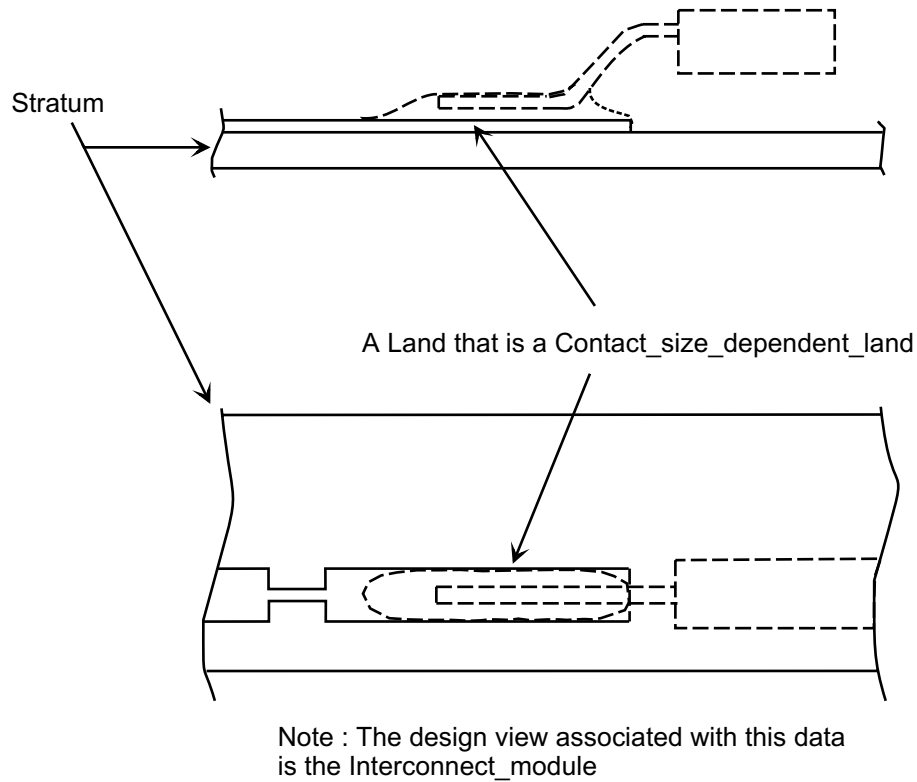


Figure 20 – Contact_size_dependent_land

NOTE EXPRESS specification:

```

*)
ENTITY contact_size_dependent_land
  SUBTYPE OF (land);
  SELF\assembly_component.usage_definition :
  default_attachment_size_based_land_physical_template;
END_ENTITY; -- contact_size_dependent_land

```

(*

4.2.195 Contact_size_dependent_non_functional_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Contact_size_dependent_non_functional_land` is a type of `Non_functional_land` (see 4.2.562) whose shape is dependent on the size of the contact area of the `Component_terminal` (see 4.2.137) that will be joined to the `Contact_size_dependent_land` (see 4.2.194) in the assembly.

The data associated with a `Contact_size_dependent_non_functional_land` are the following:

— `usage_definition`.

4.2.195.1 `usage_definition`

The `usage_definition` specifies the role of the `Default_attachment_size_based_land_physical_template` (see 4.2.247) for the `Contact_size_dependent_non_functional_land`.

NOTE EXPRESS specification:

```
*)
ENTITY contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  SELF\assembly_component.usage_definition :
  default_attachment_size_based_land_physical_template;
END_ENTITY; -- contact_size_dependent_non_functional_land

( *
```

4.2.196 `Contract`

A `Contract` is a type of `Managed_design_object` (see 4.2.524). A `Contract` is the identification of certain enterprise data related to a formal agreement between two business entities that is useful for managing product data.

NOTE This part of ISO 10303 does not support the exchange of the contents of a contract.

The data associated with a `Contract` are the following:

— `contractor`;

— `contract_approval`;

— `contract_date`;

— `contract_number`.

4.2.196.1 contractor

The contractor specifies either the Person_and_organization (see 4.2.647) or the Organization (see 4.2.569) performing work under the scope of the Contract. The contractor need not be specified for a particular Contract.

4.2.196.2 contract_approval

The contract_approval specifies the role of the Ee_approval (see 4.2.314) for the Contract. The contract_approval is optional and need not be supplied.

4.2.196.3 contract_date

The contract_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) for the Contract. The contract_date is optional and need not be supplied.

4.2.196.4 contract_number

The contract_number specifies the role of the Ee_text (see 4.2.339) for the identifier of the Contract.

Constraints:

UR1: The contract_number shall be unique within a population of Contracts.

NOTE EXPRESS specification:

```

*)
ENTITY contract
  SUBTYPE OF (managed_design_object);
    contractor      : OPTIONAL person_organization;
    contract_number  : ee_text;
    contract_approval : OPTIONAL ee_approval;
    contract_date    : OPTIONAL date_or_date_and_time;
  UNIQUE
    UR1 : contract_number;
END_ENTITY; -- contract
( *
```

4.2.197 Coordinated_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Coordinated_characteristic` is a type of `Managed_design_object` (see 4.2.524) that may include more than one discrete `Characteristic` (see 4.2.98) and that is dependent upon a specified environment for its validity. A `Coordinated_characteristic` may be a `Tolerance_characteristic` (see 4.2.911) or may be a `Range_characteristic` (see 4.2.760). The members of `component_characteristic` shall be coordinated in accordance with the definition associated with the `type_name` in the `reference_document`.

NOTE This part of ISO 10303 predefines several categories of `Tolerance_characteristic`.

The data associated with a `Coordinated_characteristic` are the following:

- `component_characteristic`;
- `dependent_environment`;
- `reference_document`;
- `type_name`.

EXAMPLE 1 An example of `coordinated_characteristic` is Poisson's ratio. This measurement, for some materials, is dependent upon the orientation of the measurement relative to the specimen being measured. Poisson's ratio, as a characteristic, is specified by three values of stress strain ratio along three orthogonal axes where one axis is along a characteristic direction of the material being described.

4.2.197.1 `component_characteristic`

The `component_characteristic` specifies the `Characteristic` (see 4.2.98) for the `Coordinated_characteristic`. There shall be one or more `Characteristic` for a `Coordinated_characteristic`.

4.2.197.2 `dependent_environment`

The `dependent_environment` specifies the text that describes the environment under which the `Coordinated_characteristic` is valid. The `dependent_environment` need not be specified for a particular `Coordinated_characteristic`.

4.2.197.3 `reference_document`

The `reference_document` specifies the `Ee_document` (see 4.2.316) or an identifier of the standard document for the type of `Coordinated_characteristic`.

4.2.197.4 `type_name`

The `type_name` specifies the string, defined in the `reference_document`, for the type of `Coordinated_characteristic`.

NOTE EXPRESS specification:

```

*)
ENTITY coordinated_characteristic
  SUPERTYPE OF (ONEOF(range_characteristic,tolerance_characteristic))
  SUBTYPE OF (managed_design_object);
  type_name          : STRING;
  component_characteristic : SET [1:?] OF characteristic;
  dependent_environment : OPTIONAL STRING;
  reference_document   : ee_document;
END_ENTITY; -- coordinated_characteristic
( *

```

4.2.198 Counterbore_passage_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Counterbore_passage_template is a type of Unsupported_passage_template (see 4.2.941). A Counterbore_passage_template is a combination of two round coaxial passages. The first passage is either a through-hole or a blind hole, and the orientation of the second passage (the counterbore) is the same as that of the first passage. The first passage is smaller than the second passage, which has a blind bottom condition that mates with the top of the first passage.

The data associated with a Counterbore_passage_template are the following:

- larger_passage;
- smaller_passage.

4.2.198.1 larger_passage

The larger_passage specifies the Blind_passage_template (see 4.2.70) for the Counterbore_passage_template. The larger_passage specifies the data that is associated with the larger opening of the Counterbore.

4.2.198.2 smaller_passage

The smaller_passage specifies the Unsupported_passage_template (see 4.2.941) for the Counterbore_passage_template. The smaller_passage specifies the data that is associated with the smaller opening of the Counterbore.

NOTE EXPRESS specification:

```

*)
ENTITY counterbore_passage_template
  SUBTYPE OF (unsupported_passage_template);
    smaller_passage : unsupported_passage_template;
    larger_passage  : blind_passage_template;
END_ENTITY; -- counterbore_passage_template

( *

```

4.2.199 Countersunk_passage_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Countersunk_passage_template is a type of Unsupported_passage_template (see 4.2.941). A Countersunk_passage_template is a combination of two round coaxial passages, one cylindrical and one tapered. The initial diameter of the tapered passage will be larger than the diameter of the cylindrical passage, and will linearly taper to be the same diameter as the cylindrical passage where the two passages meet.

The data associated with a Countersunk_passage_template are the following:

- constant_diameter_passage;
- tapered_passage.

4.2.199.1 constant_diameter_passage

The constant_diameter_passage specifies the Unsupported_passage_template (see 4.2.941) for the Countersunk_passage_template. The constant_diameter_passage specifies the data that is associated with the constant diameter opening of the Countersunk passage.

4.2.199.2 tapered_passage

The tapered_passage specifies the Blind_passage_template (see 4.2.70) for the Countersunk_passage_template. The tapered_passage specifies the data that is associated with the tapered area of the Countersunk passage.

NOTE EXPRESS specification:

```

*)
ENTITY countersunk_passage_template
  SUBTYPE OF (unsupported_passage_template);
    constant_diameter_passage : unsupported_passage_template;
    tapered_passage           : blind_passage_template;
END_ENTITY; -- countersunk_passage_template

--tt tc1 wd9 added
( *

```

4.2.200 Csg_2d_shape

A Csg_2d_shape is a type of Planar_shape (see 4.2.678).

NOTE EXPRESS specification:

```

*)
ENTITY csg_2d_shape
  SUBTYPE OF (planar_shape);
WHERE
  WR1: SIZEOF(QUERY ( c &#60;* SELF\shape_definition.element |
    'AP210_ARM.CURVE' IN TYPEOF(c)
  )) = 0;
END_ENTITY; -- csg_2d_shape

( *

```

4.2.201 Csg_primitive

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Csg_primitive is a type of Managed_design_object (see 4.2.524). A Csg_primitive is either a Right_angular_wedge (see 4.2.775), or a Right_circular_cylinder (see 4.2.776), or a Block (see 4.2.72). A Csg_primitive is used in forming a "CSG" solid model.

NOTE EXPRESS specification:

```

*)
ENTITY csg_primitive
  SUPERTYPE OF (ONEOF (block,right_circular_cylinder,right_angular_wedge))
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- csg_primitive

( *

```


4.2.202 Csg_solid

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Csg_solid is a type of Bound_volume_shape (see 4.2.76). A Csg_solid is a solid represented as a CSG model that is defined by a collection of primitive solids, combined using regularized boolean operations. The allowed operations are intersection, union and difference. Csg_solids may be used in conjunction with other solid models and boolean operations to construct larger solid models.

NOTE EXPRESS specification:

```
*)
ENTITY csg_solid
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- csg_solid

--tt tc1 wd9 modified
( *
```

4.2.203 Curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Curve is a type of Shape_item (see 4.2.818). A Curve may be a Bounding_curve (see 4.2.80) or a Closed_curve (see 4.2.104), or additionally may be a Composite_curve (see 4.2.155) or additionally may be a Trimmed_line (see 4.2.933). A Curve is the path of a point moving through its three-dimensional coordinate space or of a point moving through its two-dimensional coordinate space. A Curve shall be arc-wise connected. A Curve shall have an arc length greater than zero.

NOTE EXPRESS specification:

```
*)
ENTITY curve
  SUPERTYPE OF ((ONEOF (closed_curve,bounding_curve) ANDOR trimmed_line)
    ANDOR composite_curve)
  SUBTYPE OF (shape_item);
END_ENTITY; -- curve

( *
```

4.2.204 Curve_dimension

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Curve_dimension is a type of Single_boundary_dimension (see 4.2.827) that is measured along the entire path of the characterized Shape_element (see 4.2.810) the form of which is that of a curve.

NOTE 1 The attribute dimension_type is declared in the Dimensional_characteristic entity of which this entity is a subtype.

NOTE 2 The attribute size_of is declared in the Dimensional_size entity of which this entity is a subtype.

Constraints:

WR1: The dimension_type of the Curve_dimension shall not be "angular".

IP1: The form of the Shape_element that is specified as the size_of shall be consistent with the form of a curve.

NOTE EXPRESS specification:

```
*)  
ENTITY curve_dimension  
  SUBTYPE OF (single_boundary_dimension);  
END_ENTITY; -- curve_dimension
```

(*

4.2.205 Curve_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Curve_style is a type of Managed_design_object (see 4.2.524). A Curve_style is the definition of a set of properties that are of a geometric nature. Curve_style provides parametric data used by a processor to create physical material along a path, where the path is described outside of the style. The Curve_style algorithm is rule based and is formally described in the domain rules. The width_uncertainty represents the uncertainty in the originating CAD system, not a material processing tolerance.

The data associated with a Curve_style are the following:

- corner_style;
- curve_font;
- curve_width;
- end_extension;
- end_style;
- name.

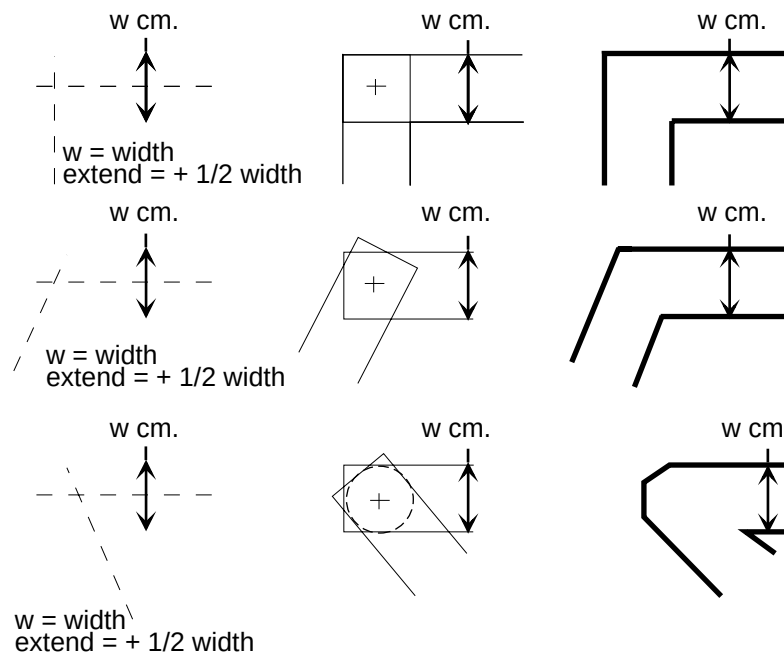


Figure 21 – Square corners with extend

4.2.205.1 corner_style

The corner_style specifies either the "extend" or "chord 2 extend" or "round" or "truncate" style for the Curve_style. Figures 21 and 23 illustrate the construction options associated with "extend", "chord 2 extend" and "truncate" corners. These illustrations shall be interpreted as describing construction of the outer boundary near the intersection only. At some distance away from the intersection, the basic width attribute governs the shape. Figure 24 illustrates the construction associated with round corners. Figure 25 illustrates the construction of the end treatment of the curve. The end treatment is applied individually to each end. Figure 26 illustrates the completed result for a Curve (see 4.2.203) that has been styled using this Curve_style. Even though the Curve does not show material inside, that is required if the Curve is applied to material.

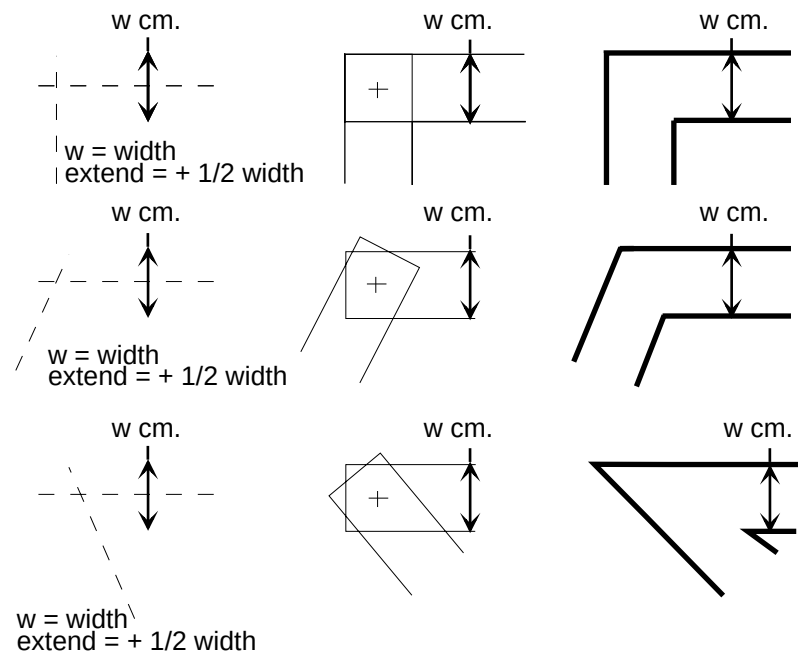


Figure 22 – Corners with positive extend

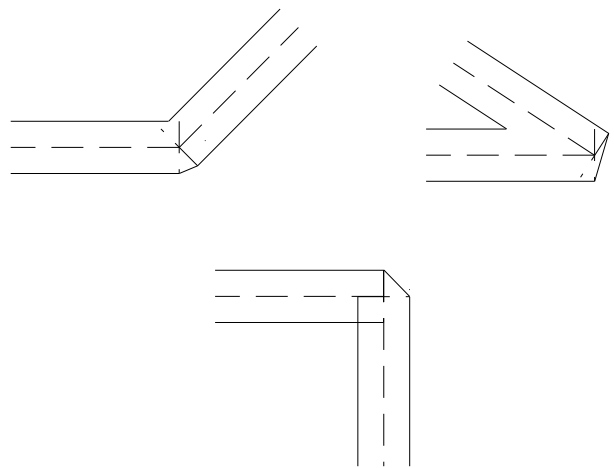


Figure 23 – Square corners with truncate

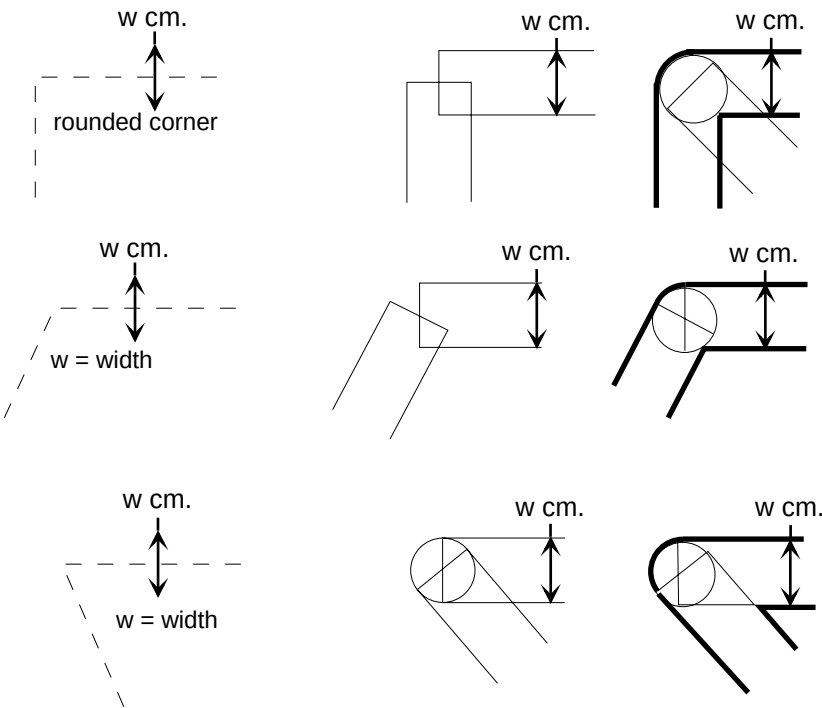


Figure 24 – Round corners

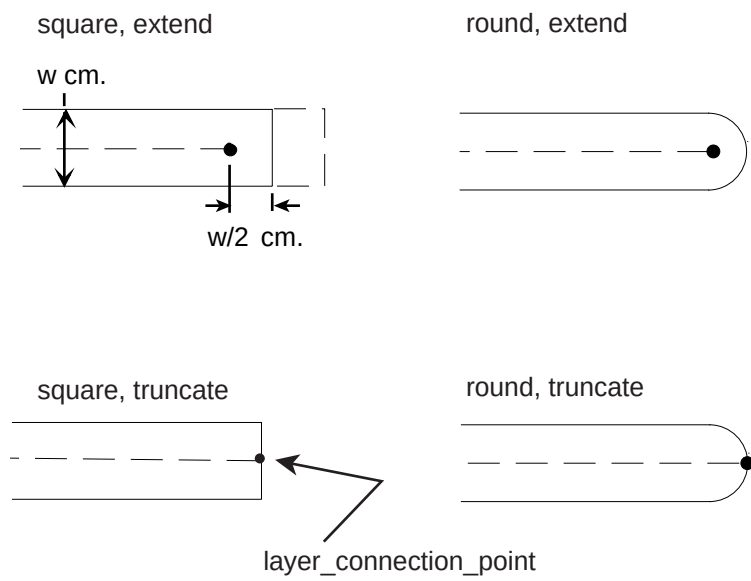
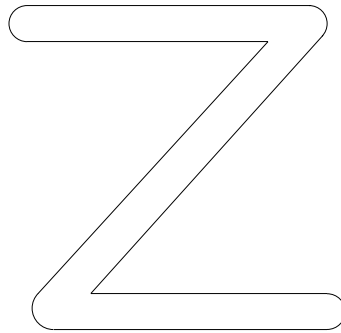


Figure 25 – Curve end treatment



Curve_style.corner_style = round
 Curve_style.end_style = round
 Curve_style.end_extension = extend

Figure 26 – Completed curve

4.2.205.2 curve_font

The curve_font specifies either the Curve_style_font_and_scaling (see 4.2.207), or one of Curve_style_font (see 4.2.206), Externally_defined_curve_font (see 4.2.362), or Solid_curve (see 4.2.831) for the Curve_style. The curve_font need not be specified.

4.2.205.3 curve_width

The curve_width specifies the role of the Length_data_element (see 4.2.511) for the Curve_style.

4.2.205.4 end_extension

The end_extension specifies how the physical material will terminate on the Layer_connection_point (see 4.2.499) that is the start or end of the centreline curve. The end_extension is applied by a continuation (termination) of the Curve (see 4.2.203) that is being styled along the tangent to the Curve at the terminus of the Curve. If the end_extension is "extend", there will be material added after the Layer_connection_point. If the end_extension is "truncate", there will be no material added after the Layer_connection_point.

NOTE This Application Object does not describe tolerances associated with fabrication processes. Reference Stratum_technology (see 4.2.871) as an example of tolerance information related to fabrication processes.

4.2.205.5 end_style

The end_style specifies either the "round" or "square" style ends for the Curve_style. Figures 21 and 23 illustrate the construction options associated with square corners. These illustrations shall be interpreted as describing construction of the outer boundary near the end only. At some distance away from the end, the basic width attribute governs the shape. Figure 25 illustrates the construction of the end treatment of the curve. The end treatment is applied individually to each end. Figure 26 illustrates the completed result for a Curve (see 4.2.203) that has been styled using this Curve_style. Even though the Curve does not show material inside, that is required if the Curve is applied to material.

4.2.205.6 name

The name specifies the string for the identifier for the Curve_style.

Constraints:

UR1: The name shall be unique within a population of Curve_style.

WR1: When the end_extension is "truncate", the Layer_connection_point for the end of interest shall be the terminus of the curve.

WR2: When the end_style is "square", the centreline of the square shall be congruent with the tangent of the centreline curve at the Layer_connection_point for the end of interest.

WR3: When the end_style is "round" and the end_extension is "extend", the Layer_connection_point for the end of interest shall be the center of a circle with radius equal to 0.5 times width.

WR4: When the end_style is "square" and the end_extension is "extend", the Layer_connection_point for the end of interest shall be the center of a square with side equal to the width.

WR5: When the corner_style is "extend", the final shape shall be the result of a union operation and a fill operation. The union operation shall be between the styled curve segments approaching the Cartesian_point of interest in the neighbourhood of the Cartesian_point. The fill operation shall maximize the material enclosed while maintaining the following conditions: The fill boundary shall be two lines that are extensions of the right hand boundary of the filled material as it traverses the corner, in the direction of the centreline curve.

WR6: When the corner_style is "chord 2 extend", the final shape shall be the result of an intersection and fill operation between the styled curve segments approaching the Cartesian_point of interest in the neighbourhood of the Cartesian_point, where there is a 2 chord approximation to a circular arc applied, which has radius equal to 0.5 times width. The final shape shall be the result of a fill operation after the union operation is performed between the styled curve segments approaching the Cartesian_point of interest in the neighbourhood of the Cartesian_point. The circular arc shall be constructed as required by the "round" corner_style rule.

WR7: When the `corner_style` is "truncate", the final shape shall be the result of an union operation followed by a fill operation. The union operation shall be between the styled curve segments approaching the `Cartesian_point` of interest in the neighbourhood of the `Cartesian_point`, where there is a bevel end applied. The final shape shall be the result of a fill operation after the union operation is performed between the styled curve segments approaching the `Cartesian_point` of interest in the neighbourhood of the `Cartesian_point`. The fill operation shall maximize the material enclosed while maintaining the following conditions: The fill boundary line shall be perpendicular to the angle bisector formed between the two Curve segments at the `Cartesian_point` of interest; the fill boundary line shall intersect the previously described result of the union operation in exactly two places; the fill boundary line shall not contain the `Cartesian_point` of interest.

WR8: When the `corner_style` is "round", the final shape shall be the result of a union operation between the styled curve segments approaching the `Cartesian_point` of interest in the neighbourhood of the `Cartesian_point`, followed by a fill operation with a circular arc of radius 0.5 times width applied. The circular arc shall be centred on the `Cartesian_point`.

WR9: When the Curve to which the `Curve_style` is applied is closed, `end_treatment` shall be ignored.

NOTE EXPRESS specification:

```

*)
ENTITY curve_style
  SUBTYPE OF (managed_design_object);
    name          : STRING;
    corner_style   : extend_or_chord_2_extend_or_truncate_or_round;
    curve_width    : length_data_element;
    curve_font     : OPTIONAL curve_font_or_scaled_curve_font_select;
    end_extension  : extend_or_truncate;
    end_style      : square_or_round;
    width_uncertainty : length_data_element;
UNIQUE
  UR1: name;
END_ENTITY; -- curve_style
(

```

4.2.206 Curve_style_font

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Curve_style_font` is a type of `Managed_design_object` (see 4.2.524). A `Curve_style_font` is the computer interpretable font used to present the Curve (see 4.2.203).

NOTE EXPRESS specification:

```

*)
ENTITY curve_style_font
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- curve_style_font

( *

```

4.2.207 Curve_style_font_and_scaling

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Curve_style_font_and_scaling is a type of Managed_design_object (see 4.2.524). A Curve_style_font_and_scaling is the combination of the computer interpretable font and the scale factor for that font used to present the Curve (see 4.2.203).

The data associated with a Curve_style_font_and_scaling are the following:

- curve_font;
- curve_font_scaling;
- name.

4.2.207.1 curve_font

The curve_font specifies either a Curve_style_font (see 4.2.206), Curve_style_font, an Externally_defined_curve_font (see 4.2.362), Externally_defined_curve_font, or a Solid_curve (see 4.2.831), Solid_curve for the Curve_style_font_and_scaling.

4.2.207.2 curve_font_scaling

The curve_font_scaling specifies the Positive_ratio_measure (see ??) used for the Curve_style_font_and_scaling.

4.2.207.3 name

The name specifies the string used for the identifier for the combination of Curve_style_font_and_scaling.

Constraints:

UR1: The name shall be unique within a population of Curve_style_font_and_scaling.

NOTE EXPRESS specification:

```

*)
ENTITY curve_style_font_and_scaling
  SUBTYPE OF (managed_design_object);
    curve_font      : curve_style_font_select;
    name            : STRING;
    curve_font_scaling : positive_ratio_measure;
UNIQUE
  UR1: name;
END_ENTITY; -- curve_style_font_and_scaling

( *

```

4.2.208 Cutout

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cutout is a type of Inter_stratum_feature (see 4.2.432). A Cutout may be a Physical_connectivity_-interrupting_cutout (see 4.2.655). A Cutout is a closed, internal shape in a substrate. A Cutout may not extend completely through the substrate. Complex Cutouts may be created by using the facilities in Assembly_group_component (see 4.2.40).

EXAMPLE 1 Figure 27 illustrates the information required to fabricate the interconnect substrate and also assemble a component onto the interconnect. The Cutout illustrated has a form that is dependent on the component installed in it. Each end of the internal conductor path shown in the figure that mates with a component terminal should be modeled as an Interconnect_module_component_terminal (see 4.2.445). During design of the interconnect substrate, these features would be considered as interface features whose location is controlled by the mating requirement. In this example the mating requirement is simplified because the component is not a connector.

EXAMPLE 2 Figure 28 illustrates a detail that shows information required to fabricate the interconnect substrate. The Cutout illustrated includes a blind passage at the base. The blind passage would be defined by a Blind_passage_template (see 4.2.70).

Constraints:

WR1: A Cutout shall not form part of the edge of the Interconnect_module.

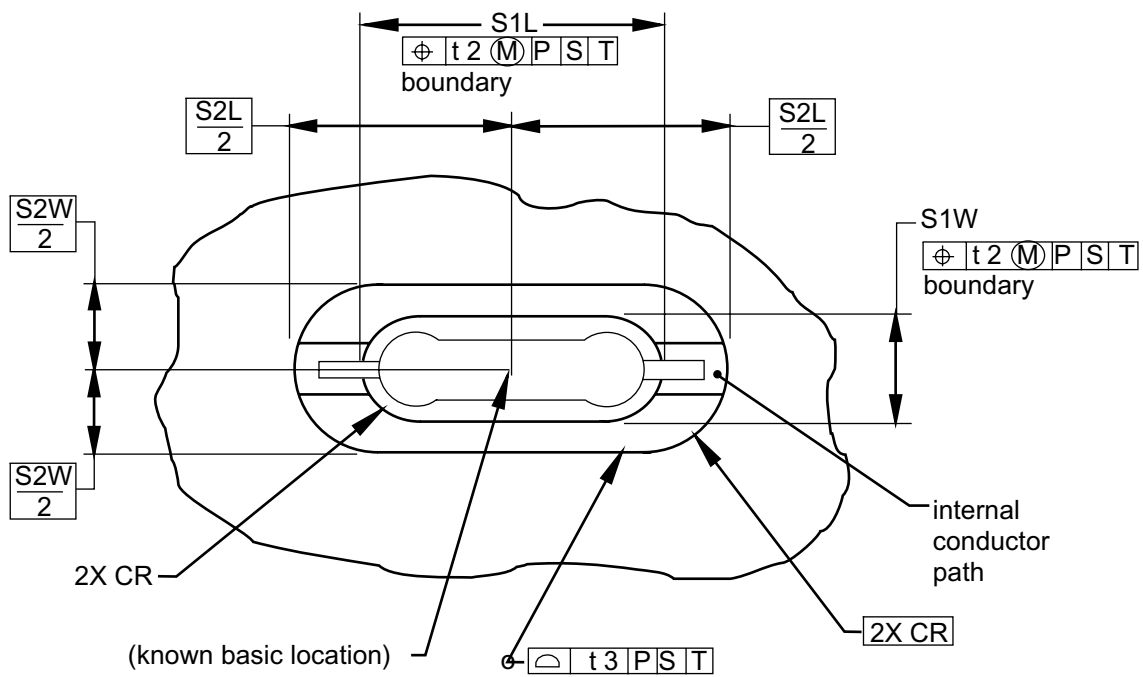


Figure 27 – Cutout plan view showing installed component

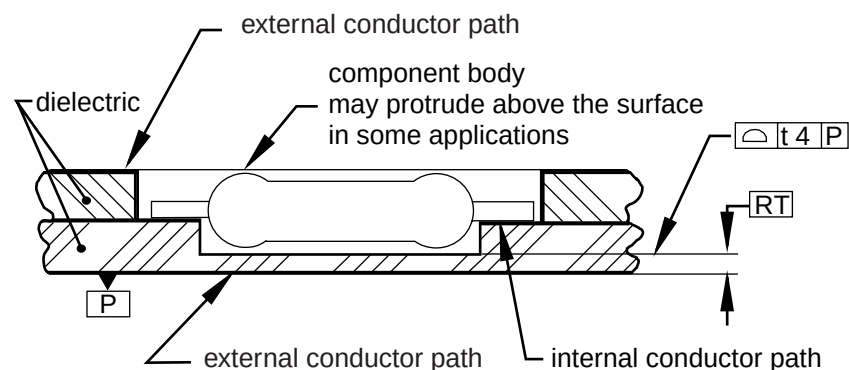


Figure 28 – Cutout cross-section view showing installed component

NOTE EXPRESS specification:

```

*)
ENTITY cutout
  SUPERTYPE OF (ONEOF(
    physical_connectivity_interrupting_cutout,
    plated_cutout, partially_plated_cutout))
  SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- cutout

--tt tc1 wd9 modified
( *
```

4.2.209 Cutout_edge_segment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cutout_edge_segment is a type of Inter_stratum_feature (see 4.2.432) that defines an edge segment of a Cutout (see 4.2.208). When the Cutout_edge_segment participates in a Cutout that is fully described by a collection of Cutout_edge_segment, the end_vertex of the final segment shall be the start_vertex of the first segment. Adjacent Cutout_edge_segment reference the same member of Edge_segment_vertex (see 4.2.312) but in different roles to ensure continuity.

EXAMPLE 1 Figure 29 illustrates a Cutout_edge_segment.

The data associated with a Cutout_edge_segment are the following:

- composed_cutout;
- end_vertex;
- start_vertex.

NOTE Some Cutouts have edges that differ in cross-section shape. This object is to describe one segment of the edge of one member of that type.

4.2.209.1 composed_cutout

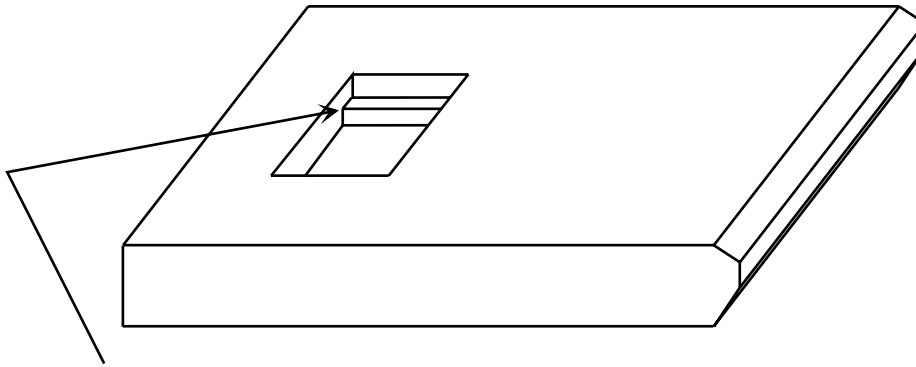
The composed_cutout specifies the Cutout (see 4.2.208) that contains the Cutout_edge_segment.

4.2.209.2 end_vertex

The end_vertex specifies one role of an Edge_segment_vertex for the Cutout_edge_segment.

4.2.209.3 start_vertex

The start_vertex specifies one role of an Edge_segment_vertex for the Cutout_edge_segment.



As with the board edge, a passage edge may have an edge section to identify an area of the passage edge that may require special treatment.

Figure 29 – Cutout_edge_segment

Constraints:

WR1: The Cutout_edge_segment shall help to compose only inner features of the Interconnect_module

WR2: The start_vertex shall not be the end_vertex.

NOTE EXPRESS specification:

```

*)
ENTITY cutout_edge_segment
  SUPERTYPE OF (plated_cutout_edge_segment)
  SUBTYPE OF (inter_stratum_feature);
    composed_cutout                : cutout;
    start_vertex                    : edge_segment_vertex;
    end_vertex                      : edge_segment_vertex;
    SELF\inter_stratum_feature.usage_definition :
inter_stratum_feature_edge_segment_template;
WHERE
  WR1: start_vertex :&#60;&#62;: end_vertex;
END_ENTITY; -- cutout_edge_segment

( *
```

4.2.210 Cylindricity_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Cylindricity_tolerance is a type of Geometric_tolerance_without_referenced_datum_system (see 4.2.423) that specifies or partially specifies the design requirement on the cylindricity of the Physical_feature_-or_part_template (see 4.2.658) to which the Cylindricity_tolerance is applied.

NOTE Cylindricity tolerance is defined in terms of a tolerance zone definition in 14.4 of ISO 1101.

Constraints:

WR1: The Shape_element specified by the toleranced_shape_element shall be a Physical_feature_or_part_template.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: The Cylindricity_tolerance shall not be a Geometric_tolerance_with_applied_material_condition_property.

WR3: The Cylindricity_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

IP1: The form of the toleranced Physical_feature_or_part_template shall be consistent with the form of a feature for which the application of a cylindricity tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY cylindricity_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- cylindricity_tolerance

--tt tc1 wd12 modified
--tt tc1 wd9 modified
(*
```

4.2.211 Data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Data_element is a type of Managed_design_object (see 4.2.524). A Data_element may be an Angle_data_element (see 4.2.23) or a Length_data_element (see 4.2.511). A Data_element is a quantitative value that includes the measurement units associated with the value. This part of ISO 10303 predefines base units from which user defined units may be mapped. ISO 10303-41 defines the mechanism for mapping between user defined units and predefined units.

The data associated with a Data_element are the following:

- measure_value;
- unit_of_measure.

4.2.211.1 measure_value

The measure_value specifies the REAL type for the Data_element. The interpretation of the measure_value shall be in accordance with the unit_of_measure.

4.2.211.2 unit_of_measure

The unit_of_measure specifies the descriptive label for the unit of measure for the associated measure_value of the Data_element. The units may be SI, derived from SI, or context dependent, and may have prefixes included whose syntax is defined in ISO 10303-41 and whose semantic is defined in ISO 31.

NOTE This note will need to turn into normative text when the xx_prefix attribute is added!!!!!!!!!!!!!!!!!!!! allowed values for prefix: (exa, peta, tera, giga, mega, kilo, hecto, deca, deci, centi, milli, micro, nano, pico, femto, atto);
 //////////////////////////////////////

NOTE EXPRESS specification:

```

*)
ENTITY data_element
  SUPERTYPE OF (ONEOF (
    absorbed_dose_data_element,
    activity_data_element,
    angle_data_element,
    area_data_element,
    capacitance_data_element,
    celsius_temperature_data_element,
    conductance_data_element,
    dose_equivalent_data_element,
    electric_charge_data_element,
    electric_current_data_element,
    electromotive_force_data_element,
    energy_data_element,
    event_data_element,
    force_data_element,
    frequency_data_element,
    illuminance_data_element,
    inductance_data_element,
    length_data_element,
    luminous_flux_data_element,
    luminous_intensity_data_element,
    magnetic_flux_data_element,
    magnetic_flux_density_data_element,
    mass_data_element,
    power_data_element,
    pressure_data_element,
    ratio_data_element,
    resistance_data_element,
    scattering_parameter_data_element,
    solid_angle_data_element,
    thermodynamic_temperature_data_element,
    time_data_element,
    volume_data_element))
  SUBTYPE OF (managed_design_object);
    measure_value    : REAL;
    unit_of_measure  : STRING;
    unit_of_measure_prefix : OPTIONAL si_prefix;
END_ENTITY; -- data_element
( *

```

4.2.212 Date

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Date is a type of Managed_design_object (see 4.2.524). A Date may be a Date_and_time (see 4.2.213). The Date is the day of the calendar.

The data associated with a Date are the following:

- day;
- month;
- year.

4.2.212.1 day

The day specifies the integer for the day of the month for the Date.

4.2.212.2 month

The month specifies the integer for the month representation for the Date.

NOTE These integers range between 1 and 12, inclusively.

4.2.212.3 year

The year specifies the integer for the year of the Date.

Constraints:

WR1: The year shall be a four digit positive integer.

NOTE EXPRESS specification:

```

*)
ENTITY date
  SUPERTYPE OF (date_and_time)
  SUBTYPE OF (managed_design_object);
  day    : INTEGER;
  year   : INTEGER;
  month  : month_representation;
END_ENTITY; -- date

( *
```

4.2.213 Date_and_time

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Date_and_time` is a type of `Date` (see 4.2.212) that includes the clock time together with the calendar Date. Resolution to the second is provided by this object.

The data associated with a `Date_and_time` are the following:

- `hour`;
- `minute`;
- `second`;
- `time_zone`.

4.2.213.1 `hour`

The `hour` specifies the integer for the period of clock time consisting of sixty minutes, for the `Date_and_time`.

4.2.213.2 `minute`

The `minute` specifies the integer for the period of clock time consisting of sixty seconds, for the `Date_and_time`.

4.2.213.3 `second`

The `second` specifies the integer for the second portion of the clock time for the `Date_and_time`.

4.2.213.4 `time_zone`

The `time_zone` specifies the integer number of hours that the local time is offset from coordinated universal time, and the direction of the offset, for the `Date_and_time`.

NOTE EXPRESS specification:

```
*)
ENTITY date_and_time
  SUBTYPE OF (date);
  hour      : INTEGER;
  minute    : INTEGER;
  second    : INTEGER;
  time_zone : INTEGER;
-- tt tc1 wd9 added
  time_zone_sense : ahead_or_behind;
END_ENTITY; -- date_and_time

( *
```

4.2.214 Datum

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum is a type of `Non_feature_shape_element` (see 4.2.560) that is a theoretically exact geometric reference such as a point, an axis, or a plane, to which tolerated `Shape_element` (see 4.2.810) are related. Each Datum is either a `Datum_point` (see 4.2.233) or a `Datum_axis` (see 4.2.215) or a `Datum_plane` (see 4.2.232). Additionally each Datum may be either a `Single_datum` (see 4.2.828) or a `Common_datum` (see 4.2.105).

EXAMPLE 1 Figure 30 illustrates the usage of a Datum.

NOTE Datums are described in clause 8 of ISO 1101.

The data associated with a Datum are the following:

— `datum_feature_usages`.

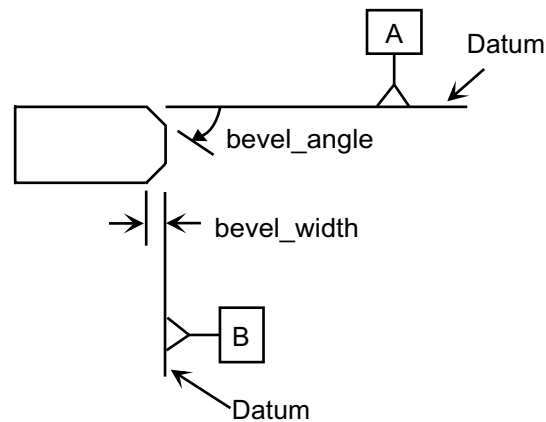


Figure 30 – Datum

4.2.214.1 datum_feature_usages

The `datum_feature_usages` is an inverse relationship that specifies that the Datum may be referenced by `Datum_feature_usage_in_datum_relationship` (see 4.2.229) that specifies the Datum as its relating_shape_element. The `datum_feature_usages` need not be specified for a particular Datum.

Constraints:

WR1: A Datum shall be a `Datum_axis`, `Datum_plane`, or `Datum_point`.

NOTE EXPRESS specification:

```

*)
ENTITY datum
  ABSTRACT SUPERTYPE OF (ONEOF (datum_point, datum_axis, datum_plane)
    ANDOR ONEOF (single_datum, common_datum))
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    datum_feature_usages : SET [0:?] OF
      datum_feature_usage_in_datum_relationship FOR
      relating_shape_element;
END_ENTITY; -- datum

```

(*

4.2.215 Datum_axis

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_axis is a type of Datum (see 4.2.214) that is a theoretically exact axis to which tolerated Shape_element (see 4.2.810) are related.

Constraints:

WR1: If the Datum_axis is not specified as a reference_axes by a Datum_reference_frame then the Datum_axis shall be either a Common_datum, or a Single_datum.

NOTE EXPRESS specification:

```

*)
ENTITY datum_axis
  SUBTYPE OF (datum);
END_ENTITY; -- datum_axis

```

(*

4.2.216 Datum_axis_and_tolerance_zone_orienting_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_axis_and_tolerance_zone_orienting_relationship is a type of Managed_design_object (see 4.2.524).
 A Datum_axis_and_tolerance_zone_orienting_relationship is a relationship that associates a Tolerance_zone (see 4.2.913) with the Datum_axis (see 4.2.215) by which it is oriented.

The data associated with a Datum_axis_and_tolerance_zone_orienting_relationship are the following:

- oriented_tolerance_zone;
- orienting_datum_axis;
- zone_orientation.

4.2.216.1 oriented_tolerance_zone

The oriented_tolerance_zone specifies the Tolerance_zone (see 4.2.913) that is oriented with respect to the Datum_axis (see 4.2.215) specified as the orienting_datum_axis.

4.2.216.2 orienting_datum_axis

The orienting_datum_axis specifies the Datum_axis (see 4.2.215) by which the Tolerance_zone (see 4.2.913) that is specified as the oriented_tolerance_zone is oriented.

4.2.216.3 zone_orientation

The zone_orientation is an inverse relationship that specifies that the existence of the Datum_axis_and_tolerance_zone_orienting_relationship is dependent on the existence of the Datum_axis_related_orientation (see 4.2.217) that specifies the Datum_axis_and_tolerance_zone_orienting_relationship as its orienting_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY datum_axis_and_tolerance_zone_orienting_relationship
  SUBTYPE OF (managed_design_object);
    oriented_tolerance_zone : tolerance_zone;
    orienting_datum_axis    : datum_axis;
  INVERSE
    zone_orientation : datum_axis_related_orientation FOR
                      orienting_relationship;
END_ENTITY; -- datum_axis_and_tolerance_zone_orienting_relationship

( *
```

4.2.217 Datum_axis_related_orientation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_axis_related_orientation is a type of Managed_design_object (see 4.2.524). A Datum_axis_related_orientation is a definition that characterizes the orientation of a Tolerance_zone (see 4.2.913) in relation to a Datum_axis (see 4.2.215).

The data associated with a Datum_axis_related_orientation are the following:

- orientation_type;
- orienting_relationship.

4.2.217.1 orientation_type

The orientation_type specifies a label that indicates the manner in which the specified Tolerance_zone (see 4.2.913) is oriented with respect to the specified Datum_axis (see 4.2.215). The orientation_type shall either be "centred_on_arc", or "centred_on_radial".

EXAMPLE 1 In the polar coordinate method of bidirectional positional tolerancing the bidirectional tolerance zone consists of two tolerance zones. One that is bounded by a pair of parallel planes that is centred along a radial from the datum axis and another that is bounded by a pair of concentric cylindrical arc boundaries that are centred along a cylindrical arc the origin of which is the datum axis. An orientation_type of "centred_on_radial" corresponds to the first tolerance zone and "centred_on_arc" corresponds to the second tolerance zone.

NOTE The polar method of bidirectional positional tolerancing is discussed in 5.9.2 of ASME Y14.5M.

4.2.217.2 orienting_relationship

The orienting_relationship specifies the Datum_axis_and_tolerance_zone_orienting_relationship (see 4.2.216) that associates the Tolerance_zone (see 4.2.913) with the Datum_axis (see 4.2.215) with which it is oriented. The specified Datum_axis_and_tolerance_zone_orienting_relationship is characterized by the Datum_axis_related_orientation.

Constraints:

WR1: The Datum_axis_related_orientation shall be specified as the zone_orientation by exactly one Tolerance_zone_definition_with_specified_orientation.

NOTE EXPRESS specification:

```
*)
ENTITY datum_axis_related_orientation
  SUBTYPE OF (managed_design_object);
    orienting_relationship : datum_axis_and_tolerance_zone_orienting_relationship;
    orientation_type       : centre_type;
END_ENTITY; -- datum_axis_related_orientation

( *
```

4.2.218 Datum_based_direction_vector

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_based_direction_vector is a type of Managed_design_object (see 4.2.524). A Datum_based_direction_vector is a characterization of a direction away from and normal to a Datum_plane (see 4.2.232).

The data associated with a Datum_based_direction_vector are the following:

- direction_vector_representation;
- positive_direction_vector;
- vector_of.

4.2.218.1 direction_vector_representation

The direction_vector_representation specifies an Axis_placement (see 4.2.56) that is a representation for the Datum_based_direction_vector.

4.2.218.2 positive_direction_vector

The positive_direction_vector specifies a boolean value that indicates whether the Datum_based_direction_vector is in a positive or negative sense. A value of TRUE indicates that the Datum_based_direction_vector is in the positive sense. A value of FALSE indicates that the Datum_based_direction_vector is in the negative sense.

4.2.218.3 vector_of

The vector_of specifies the Datum_plane (see 4.2.232) from which the Datum_based_direction_vector is based.

Constraints:

WR1: The Datum_based_direction_vector shall be specified as the direction by at least one Direction_element.

NOTE EXPRESS specification:

```

*)
ENTITY datum_based_direction_vector
  SUBTYPE OF (managed_design_object);
    direction_vector_representation : axis_placement;
    vector_of                       : datum_plane;
    positive_direction_vector       : BOOLEAN;
END_ENTITY; -- datum_based_direction_vector

( *

```

4.2.219 Datum_based_length_measure

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_based_length_measure is a type of Managed_design_object (see 4.2.524). A Datum_based_length_measure is a dimensional measure of length that is taken from a defined Datum (see 4.2.214).

EXAMPLE 1 Figure 31 illustrates the usage of a Datum_based_length_measure.

The data associated with a Datum_based_length_measure are the following:

- measure;
- measure_orientation;
- of_datum.

4.2.219.1 measure

The measure specifies the role of the Length_data_element (see 4.2.511) for the Datum_based_length_measure.

4.2.219.2 measure_orientation

The measure_orientation specifies the role of the Direction (see 4.2.304) for the Datum_based_length_measure or specifies the role of the string 'normal' or specifies the role of the string 'reversed'. 'Normal' shall be interpreted as a description equivalent to the three dimensional concept of a vector parallel to the normal of the plane that is the datum upon which the Datum_based_length_measure is based. 'Reversed' shall be interpreted as a description equivalent to the three dimensional concept of a vector anti-parallel to the normal of the plane that is the datum upon which the Datum_based_length_measure is based. For the case that the Datum is a Datum_plane, the Direction vector shall be in the XZ plane of the Datum_plane, where the Z axis is the normal vector of the Datum_plane. The measure_orientation is measured from the X axis in the XZ plane. For the two-dimensional case, the Datum_plane if used, is considered to be congruent with the XY coordinate system.

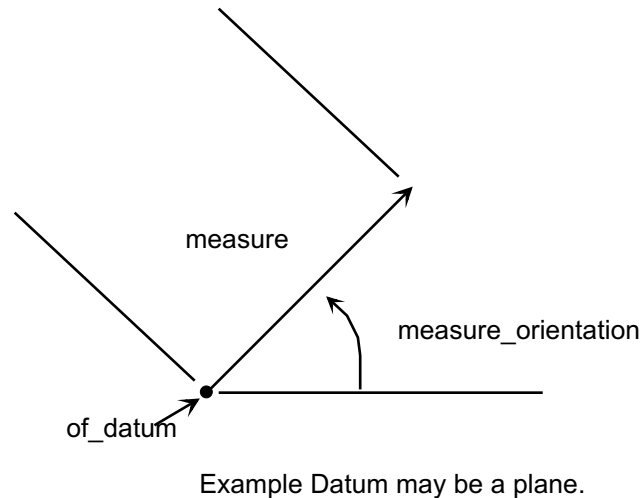


Figure 31 – Datum_based_length_measure

NOTE The use of descriptive items allows the information to be conveyed when the geometric context is 2d, since orientation only supports a description within a 3d geometric context.

4.2.219.3 of_datum

The of_datum specifies the role of the Datum_plane (see 4.2.232) for the Datum_based_length_measure. A Datum_plane is required in order to support equivalent description between the 2d and 3d shape representation cases.

NOTE EXPRESS specification:

```

*)
ENTITY datum_based_length_measure
  SUBTYPE OF (managed_design_object);
    measure          : length_data_element;
    measure_orientation : descriptive_or_geometric_orientation;
    of_datum         : datum_plane;
END_ENTITY; -- datum_based_length_measure

( *
```

4.2.220 Datum_based_vector_orientation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_based_vector_orientation is a type of Managed_design_object (see 4.2.524). A Datum_based_vector_orientation is a definition of the orientation of a shape element that has the form of a vector and is based on the intersection of two shape elements that have the form of half-planes. Where a half-plane has the form of a plane except it is bounded on one side by a straight line. Each half-plane shape element is oriented with respect to three Datum_plane (see 4.2.232) and is defined by a Direction_element (see 4.2.305).

The data associated with a Datum_based_vector_orientation are the following:

— direction_components.

4.2.220.1 direction_components

The direction_components specifies the Direction_element (see 4.2.305) that defines the half-plane shape elements the intersection of which forms the vector that is defined by the Datum_based_vector_orientation. There shall be exactly two Direction_element for the Datum_based_vector_orientation.

Constraints:

WR1: The Datum_based_vector_orientation shall be specified as the zone_orientation by at least one Tolerance_zone_definition_with_specified_orientation, or as the measurement_direction by at least one Dimension_location_with_direction_vector, or as the measurement_orientation by at least one Angular_dimension_with_direction_vector.

WR2: The Datum_planes specified as the base_datum_plane by the two Direction_element specified as direction_components shall be different.

WR3: The Datum_planes specified as the base_datum_plane by the two Direction_element specified as direction_components shall be Datum_planes of the same Datum_reference_frame.

NOTE EXPRESS specification:

```
*)
ENTITY datum_based_vector_orientation
  SUBTYPE OF (managed_design_object);
  direction_components : SET [2:2] OF direction_element;
END_ENTITY; -- datum_based_vector_orientation
```

(*

4.2.221 Datum_difference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_difference is a type of Composite_unit_shape_element (see 4.2.167). A Datum_difference is the basis for mapping from a geometric context model into a lexical context model for definitional purposes. A Datum_difference identifies the length or angular difference between two Datum. The difference may be referenced independently of the originating geometric model representation.

NOTE 1 One Datum may be located at the origin if a simple location is intended.

NOTE 2 The origin reference for a simple angular difference should be the X axis for two-dimensional case, and the XY axis for three-dimensional case.

The data associated with a Datum_difference are the following:

- precedent_datum;
- subsequent_datum.

4.2.221.1 precedent_datum

A precedent_datum specifies one role of a Datum (see 4.2.214) for a Datum_difference. The precedent_datum is the reference datum.

4.2.221.2 subsequent_datum

A subsequent_datum specifies one role of a Datum (see 4.2.214) for a Datum_difference. The subsequent_datum is the measureand.

NOTE EXPRESS specification:

```

*)
ENTITY datum_difference
  SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.name : label;
DERIVE
  precedent_datum :
datum := SELF\shape_element_relationship.relatng_shape_element;
  subsequent_datum :
datum := SELF\shape_element_relationship.related_shape_element;
END_ENTITY; -- datum_difference

( *
```

4.2.222 Datum_difference_analytical_model_port_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_difference_analytical_model_port_assignment is a type of Managed_design_object (see 4.2.524). A Datum_difference_analytical_model_port_assignment is provided to map from the geometric context into a specific Analytical_model through its ports. The use of Datum_difference (see 4.2.221) isolates the model ports from a specific geometric representation.

The data associated with a Datum_difference_analytical_model_port_assignment are the following:

- assigned_analytical_model_port;
- of_datum_difference.

4.2.222.1 assigned_analytical_model_port

An assigned_analytical_model_port specifies the role of the Analytical_model_port (see 4.2.17) for the Datum_difference_analytical_model_port_assignment.

4.2.222.2 of_datum_difference

An of_datum_difference specifies the role of the Datum_difference (see 4.2.221) for the Datum_difference_analytical_model_port_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY datum_difference_analytical_model_port_assignment
  SUBTYPE OF (managed_design_object);
    assigned_analytical_model_port : analytical_model_port;
    of_datum_difference             : datum_difference;

END_ENTITY; -- datum_difference_analytical_model_port_assignment

( *
```

4.2.223 Datum_difference_based_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_difference_based_characteristic is a type of Characteristic (see 4.2.98). A Datum_difference_based_characteristic is provided to allow characterization based on a geometric model. This is typically used where there is a reference geometric model available.

The data associated with a Datum_difference_based_characteristic are the following:

- of_datum_difference.

4.2.223.1 of_datum_difference

An of_datum_difference specifies the role of the Datum_difference (see 4.2.221) for the Datum_difference_based_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY datum_difference_based_characteristic
  SUBTYPE OF (characteristic);
    of_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_based_characteristic

( *
```

4.2.224 Datum_difference_based_model_parameter

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_difference_based_model_parameter is a type of Model_parameter (see 4.2.549). In effect, the Datum_difference provides a geometric basis for the definition of the Model_parameter.

The data associated with a Datum_difference_based_model_parameter are the following:

— of_datum_difference.

4.2.224.1 of_datum_difference

An of_datum_difference specifies the role of the Datum_difference (see 4.2.221) for the Datum_difference_based_model_parameter.

NOTE EXPRESS specification:

```
*)
ENTITY datum_difference_based_model_parameter
  SUBTYPE OF (model_parameter);
    of_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_based_model_parameter

( *
```

4.2.225 Datum_difference_functional_unit_usage_view_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_difference_functional_unit_usage_view_terminal_assignment is a type of Managed_design_object (see 4.2.524). A Datum_difference_functional_unit_usage_view_terminal_assignment provides the ability to map from a geometric context into the functional context of a Functional_unit_definition (see 4.2.401).

The data associated with a Datum_difference_functional_unit_usage_view_terminal_assignment are the following:

- assigned_functional_terminal_definition;
- assigned_datum_difference.

4.2.225.1 assigned_functional_terminal_definition

An assigned_functional_terminal_definition specifies the role of the Functional_unit_usage_view_terminal_definition (see 4.2.411) for the Datum_difference_functional_unit_usage_view_terminal_assignment.

4.2.225.2 assigned_datum_difference

An assigned_datum_difference specifies the role of the Datum_difference (see 4.2.221) for the Datum_difference_functional_unit_usage_view_terminal_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY datum_difference_functional_unit_usage_view_terminal_assignment
  SUBTYPE OF (managed_design_object);
    assigned_functional_terminal_definition :
functional_unit_usage_view_terminal_definition;
    assigned_datum_difference                : datum_difference;
END_ENTITY; -- datum_difference_functional_unit_usage_view_terminal_assignment

( *
```

4.2.226 Datum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_feature is a type of Physical_feature_or_part_template (see 4.2.658) that is used to establish one or more Datum (see 4.2.214). A Datum_feature may be a Datum_target_set (see 4.2.241). A Datum_feature may be used to establish one Single_datum (see 4.2.828), and one or more Common_datum (see 4.2.105).

NOTE The definition of datum feature is given in 3.3 of ISO 5459.

The data associated with a Datum_feature are the following:

- datum_feature_usages;
- identification.

4.2.226.1 datum_feature_usages

The datum_feature_usages is an inverse relationship that specifies that the existence of the Datum_feature is dependent on the existence of the Datum_feature_usage_in_datum_relationship (see 4.2.229) that specifies the Datum_feature as its related_shape_element. There shall be one or more Datum_feature_usage_in_datum_relationship for a Datum_feature.

4.2.226.2 identification

The identification specifies the label by which the Datum_feature is referred.

NOTE Datum (see 4.2.214) letter identifiers are examples of identifiers of Datum features. The Datum letter identifier is described in 6.1.2 of ISO 5459.

Constraints:

WR1: A Datum_feature shall be used to establish at most one Single_datum.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature
  SUPERTYPE OF (datum_target_set)
  SUBTYPE OF (physical_feature_or_part_template);
  identification : label;
  INVERSE
    datum_feature_usages : SET [1:?] OF
      datum_feature_usage_in_datum_relationship FOR
      related_shape_element;
END_ENTITY; -- datum_feature

( *
```

4.2.227 Datum_feature_material_condition_property

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Datum_feature_material_condition_property` is a type of `Managed_design_object` (see 4.2.524). A `Datum_feature_material_condition_property` is a specification of a material condition property that defines the manner in which `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422) requirements are dependent on the produced size of a `Datum_feature` (see 4.2.226) of the `Datum_system` (see 4.2.236).

The data associated with a `Datum_feature_material_condition_property` are the following:

- `applied_to`;
- `datum_system_conditions_definition`;
- `name`.

4.2.227.1 `applied_to`

The `applied_to` specifies the `Datum_feature_usage_in_datum_system_relationship` (see 4.2.230) that identifies the usage of a `Datum_feature` (see 4.2.226) in a `Datum_system` (see 4.2.236) and to which the material condition property is applied.

4.2.227.2 `datum_system_conditions_definition`

The `datum_system_conditions_definition` is an inverse relationship that specifies that the existence of the `Datum_feature_material_condition_property` depends on the existence of the `Datum_system_defining_conditions_with_material_conditions` (see 4.2.238) that specifies the `Datum_feature_material_condition_property` as its `applied_material_condition_properties`.

4.2.227.3 `name`

The `name` specifies the type of the applied material condition property, either "`least_material_requirement`" or "`maximum_material_principle`" for the `Datum_feature_material_condition_property`.

NOTE 1 The application of the maximum material principle to a Datum (see 4.2.214) feature is described in 4.3 ISO 2692.

NOTE 2 The application of the least material requirement to a Datum feature is described in clause 3 of ISO 2692-Amendment 1.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature_material_condition_property
  SUBTYPE OF (managed_design_object);
    applied_to : datum_feature_usage_in_datum_system_relationship;
    name       : material_condition_property_names;
  INVERSE
    datum_system_conditions_definition :
datum_system_defining_conditions_with_material_conditions FOR
applied_material_condition_properties;
END_ENTITY; -- datum_feature_material_condition_property

```

(*

4.2.228 Datum_feature_usage_in_common_datum_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_feature_usage_in_common_datum_relationship is a type of Datum_feature_usage_in_datum_relationship (see 4.2.229) that associates a Datum_feature (see 4.2.226) with a Common_datum (see 4.2.105) that the Datum_feature participates in establishing.

The data associated with a Datum_feature_usage_in_common_datum_relationship are the following:

— relating_shape_element.

4.2.228.1 relating_shape_element

The relating_shape_element specifies the Common_datum (see 4.2.105) that the Datum_feature (see 4.2.226) specified as the related_shape_element participates in establishing for the Datum_feature_usage_in_common_datum_relationship.

NOTE The attribute related_shape_element is declared in the Datum_feature_usage_in_datum_relationship (see 4.2.229) entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature_usage_in_common_datum_relationship
  SUBTYPE OF (datum_feature_usage_in_datum_relationship);
    SELF\shape_element_relationship.relatating_shape_element : common_datum;
END_ENTITY; -- datum_feature_usage_in_common_datum_relationship

```

(*

4.2.229 Datum_feature_usage_in_datum_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_feature_usage_in_datum_relationship is a type of Shape_element_relationship (see 4.2.816) that associates a Datum_feature (see 4.2.226) with a Datum (see 4.2.214) that the Datum_feature participates in establishing. A Datum_feature_usage_in_datum_relationship is either a Datum_feature_usage_in_common_datum_relationship (see 4.2.228) or a Datum_feature_usage_in_single_datum_relationship (see 4.2.231).

The data associated with a Datum_feature_usage_in_datum_relationship are the following:

- related_shape_element;
- relating_shape_element.

4.2.229.1 related_shape_element

The related_shape_element specifies the Datum_feature (see 4.2.226) that is used to establish the Datum (see 4.2.214) specified as the relating_shape_element for the Datum_feature_usage_in_datum_relationship.

4.2.229.2 relating_shape_element

The relating_shape_element specifies the Datum (see 4.2.214) that the specified Datum_feature (see 4.2.226) participates in establishing for the Datum_feature_usage_in_datum_relationship.

Constraints:

UR1: The combination of relating_shape_element and related_shape_element shall be unique within a population of Datum_feature_usage_in_datum_relationship. That is, a specific Datum_feature shall not be used more than once to establish a specific Datum.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature_usage_in_datum_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (
    datum_feature_usage_in_single_datum_relationship,
    datum_feature_usage_in_common_datum_relationship))
  SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.relatng_shape_element : datum;
  SELF\shape_element_relationship.related_shape_element : datum_feature;
  UNIQUE
  UR1 : related_shape_element, relating_shape_element;
END_ENTITY; -- datum_feature_usage_in_datum_relationship

( *

```

4.2.230 Datum_feature_usage_in_datum_system_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_feature_usage_in_datum_system_relationship is a type of Managed_design_object (see 4.2.524). A Datum_feature_usage_in_datum_system_relationship is a relationship that associates a Datum_feature (see 4.2.226) with a Datum_system (see 4.2.236) that the Datum_feature participates in establishing.

The data associated with a Datum_feature_usage_in_datum_system_relationship are the following:

- applied_material_condition_property;
- established_datum_system;
- used_datum_feature.

4.2.230.1 applied_material_condition_property

The applied_material_condition_property is an inverse relationship that specifies that the existence of the Datum_feature_usage_in_datum_system_relationship depends on the existence of the Datum_feature_material_condition_property (see 4.2.227) that specifies the Datum_feature_usage_in_datum_system_relationship as its applied_to.

4.2.230.2 established_datum_system

The established_datum_system specifies the Datum_system (see 4.2.236) in which the specified Datum_feature (see 4.2.226) participates in establishing for the Datum_feature_usage_in_datum_system_relationship.

4.2.230.3 used_datum_feature

The used_datum_feature specifies the Datum_feature (see 4.2.226) that is used to establish the specified Datum_system (see 4.2.236) for the Datum_feature_usage_in_datum_system_relationship.

Constraints:

WR1: The Datum_feature specified by the used_datum_feature shall establish a Datum that participates in the establishment of the Datum_system specified as the established_datum_system.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature_usage_in_datum_system_relationship
  SUBTYPE OF (managed_design_object);
    established_datum_system : datum_system;
    used_datum_feature       : datum_feature;
  INVERSE
    applied_material_condition_property :
datum_feature_material_condition_property FOR applied_to;
END_ENTITY; -- datum_feature_usage_in_datum_system_relationship

( *
```

4.2.231 Datum_feature_usage_in_single_datum_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_feature_usage_in_single_datum_relationship is a type of Datum_feature_usage_in_datum_relationship (see 4.2.229) that associates a Datum_feature (see 4.2.226) with the Single_datum (see 4.2.828) that the Datum_feature establishes.

The data associated with a Datum_feature_usage_in_single_datum_relationship are the following:

— relating_shape_element.

4.2.231.1 relating_shape_element

The relating_shape_element specifies the Single_datum (see 4.2.828) that the Datum_feature (see 4.2.226) specified as the related_shape_element establishes for the Datum_feature_usage_in_single_datum_relationship.

NOTE The attribute related_shape_element is declared in the Datum_feature_usage_in_datum_relationship entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY datum_feature_usage_in_single_datum_relationship
  SUBTYPE OF (datum_feature_usage_in_datum_relationship);
  SELF\shape_element_relationship.relate_shape_element : single_datum;
END_ENTITY; -- datum_feature_usage_in_single_datum_relationship

( *

```

4.2.232 Datum_plane

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_plane is a type of Datum (see 4.2.214) that is a theoretically exact plane to which tolerated Shape_element (see 4.2.810) are related.

Constraints:

WR1: If the Datum_plane is not specified as a reference_planes of a Datum_reference_frame then the Datum_plane shall be either a Common_datum, or a Single_datum.

WR2: A Datum_plane shall be specified as the vector_of by at most two Datum_based_direction_vector.

WR3: If the Datum_plane is specified as the vector_of by two Datum_based_direction_vector, the two Datum_based_direction_vector shall specify different values for the positive_direction_vector.

NOTE EXPRESS specification:

```

*)
ENTITY datum_plane
  SUBTYPE OF (datum);
END_ENTITY; -- datum_plane

--tt tc1 wd9 modified
( *

```

4.2.233 Datum_point

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_point is a type of Datum (see 4.2.214) that is a theoretically exact point to which tolerated Shape_element (see 4.2.810) are related.

Constraints:

WR1: If the Datum_point is not specified as a reference_origin of a Datum_reference_frame then the Datum_point shall be either a Common_datum, or a Single_datum.

NOTE EXPRESS specification:

```

*)
ENTITY datum_point
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (datum);
  location_in_2d : OPTIONAL cartesian_point;
  location_in_3d : OPTIONAL cartesian_point;
  associated_volume_shape_intersection : OPTIONAL volume_shape_intersection;
WHERE
  WR1: NOT(EXISTS(location_in_2d) AND EXISTS(location_in_3d)) OR
        EXISTS(associated_volume_shape_intersection);
  WR2: NOT(EXISTS(location_in_2d)) OR (location_in_2d.geometric_context.dimensionality
                                     = two_dimensional);
  WR3: NOT(EXISTS(location_in_3d)) OR (location_in_3d.geometric_context.dimensionality
                                     = three_dimensional);
END_ENTITY; -- datum_point

( *
```

4.2.234 Datum_precedence_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_precedence_assignment is a type of Managed_design_object (see 4.2.524). A Datum_precedence_assignment is a specification that defines the order in which a Datum (see 4.2.214) is used to establish a Datum_system (see 4.2.236).

NOTE Datum systems established by a sequence of datums is discussed in 6.2.3 of ISO 5459.

The data associated with a Datum_precedence_assignment are the following:

- assigned_to;
- datum_system_conditions_definition;
- name.

4.2.234.1 assigned_to

The assigned_to specifies the Datum_usage_in_datum_system_relationship (see 4.2.243) that associates a Datum_system (see 4.2.236) with the Datum (see 4.2.214) to which the precedence in establishing the Datum_system is assigned by the Datum_precedence_assignment.

4.2.234.2 datum_system_conditions_definition

The datum_system_conditions_definition is an inverse relationship that specifies that the existence of the Datum_precedence_assignment depends on the existence of the Datum_system_defining_conditions (see 4.2.237) that specifies the Datum_precedence_assignment as its assigned_datum_precedences.

4.2.234.3 name

The name specifies the label that corresponds to the Datum_precedence_assignment of the specified usage of a Datum (see 4.2.214) in a Datum_system (see 4.2.236). The name shall be "primary", "secondary", or "tertiary".

EXAMPLE 1 The requirements may be such that Datum A is assigned as the primary Datum when it is used in one Datum_system. However, additional requirements may be such that Datum A is assigned as the tertiary Datum when it is used in a second Datum_system.

Constraints:

UR1: The combination of name and datum_system_conditions_definition shall be unique within a population of Datum_precedence_assignment. That is, the defining conditions of a Datum_system shall be such that no two Datum of the Datum_system have been assigned the same precedence.

NOTE EXPRESS specification:

```

*)
ENTITY datum_precedence_assignment
  SUBTYPE OF (managed_design_object);
    assigned_to : datum_usage_in_datum_system_relationship;
    name       : precedence_type;
  INVERSE
    datum_system_conditions_definition : datum_system_defining_conditions FOR
                                         assigned_datum_precedences;
  UNIQUE
    UR1 : datum_system_conditions_definition, name;
END_ENTITY; -- datum_precedence_assignment

( *
```

4.2.235 Datum_reference_frame

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_reference_frame is a type of Non_feature_shape_element (see 4.2.560) that is a system of Datum (see 4.2.214) that serve as a frame of reference by which tolerated Shape_element (see 4.2.810) may be located and oriented. The Datum_reference_frame is established by one or more Datum_system (see 4.2.236). The Datum_reference_frame consists of three mutually perpendicular Datum_plane (see 4.2.232), three mutually perpendicular Datum_axis (see 4.2.215), and a Datum_point (see 4.2.233). Each of the Datum_axis is located at the intersections of each unique pair of Datum_plane. The Datum_point is the origin of the Datum_reference_frame and is located at the mutual intersection of the three Datum_plane.

The data associated with a Datum_reference_frame are the following:

- reference_axes;
- reference_axis_pairs;
- reference_origin;
- reference_planes;
- reference_plane_pairs;
- used_datum_systems.

4.2.235.1 reference_axes

The reference_axes specifies the Datum_axis (see 4.2.215) that are part of the Datum_reference_frame. There shall be three Datum_axis specified for a Datum_reference_frame.

4.2.235.2 reference_axis_pairs

The reference_axis_pairs specifies the Perpendicular_shape_element_relationship (see 4.2.644) that define the perpendicularity between the reference_axes. There shall be three Perpendicular_shape_element_relationship in this role specified for a Datum_reference_frame.

4.2.235.3 reference_origin

The reference_origin specifies the Datum_point (see 4.2.233) that corresponds to the origin of the Datum_reference_frame.

4.2.235.4 reference_planes

The reference_planes specify the Datum_plane (see 4.2.232) that are part of the Datum_reference_frame. There shall be three Datum_plane specified for a Datum_reference_frame.

4.2.235.5 reference_plane_pairs

The `reference_plane_pairs` specifies the `Perpendicular_shape_element_relationship` (see 4.2.644) that define the perpendicularity between the reference_planes. There shall be three `Perpendicular_shape_element_relationship` in this role specified for a `Datum_reference_frame`.

4.2.235.6 used_datum_systems

The `used_datum_systems` is an inverse relationship that specifies that the existence of the `Datum_reference_frame` is dependent on the existence of the `Datum_system` (see 4.2.236) that specifies the `Datum_reference_frame` as its established_datum_reference_frame. There shall be one or more `Datum_system` for the `Datum_reference_frame`. Only members of `reference_axes` or `reference_planes` shall be `Datum` of the `used_datum_systems`.

Constraints:

WR1: Each of the `reference_axes` shall be specified to be an intersection of one of the pairs of `reference_planes`.

NOTE EXPRESS specification:

*)

```
ENTITY datum_reference_frame
  SUBTYPE OF (non_feature_shape_element);
    reference_planes      : SET [3:3] OF datum_plane;
    reference_plane_pairs : SET [3:3] OF perpendicular_shape_element_relationship;
    reference_axes        : SET [3:3] OF datum_axis;
    reference_axis_pairs  : SET [3:3] OF perpendicular_shape_element_relationship;
    reference_origin      : datum_point;
  INVERSE
    used_datum_systems : SET [1:?] OF datum_system FOR
                        established_datum_reference_frame;
```

WHERE

```
  WR1: 'AP210_ARM.DERIVED_SHAPE_ELEMENT' IN TYPEOF(reference_origin);
```

(*

```
  WR2: reference_planes_referenced_by_reference_plane_pairs(reference_planes, reference_planes);
```

```
  WR3: reference_axes_referenced_by_reference_axis_pairs(reference_axes, reference_axes);
```

```
  WR4: valid_reference_axis_reference_plane_relationship(reference_axes, reference_planes);
```

```
  WR5: valid_reference_origin_derivation_from_planes(reference_origin, reference_planes);
```

*)

```
  WR6: SIZEOF(QUERY( ra &#60;* reference_axes | NOT
    ('AP210_ARM.DERIVED_SHAPE_ELEMENT' IN TYPEOF(ra))
  )) = 0;
```

```
END_ENTITY; -- datum_reference_frame
```

(*

4.2.236 Datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_system is a type of Non_feature_shape_element (see 4.2.560) that is a system of Datum (see 4.2.214) that establish a Datum_reference_frame (see 4.2.235) by which toleranced Shape_element (see 4.2.810) may be located and oriented.

EXAMPLE 1 Figure 32 illustrates a Datum_system.

NOTE The definition of datum-system is given in 3.2 of ISO 5459.

The data associated with a Datum_system are the following:

- datum_usages;
- defining_conditions;
- established_datum_reference_frame.

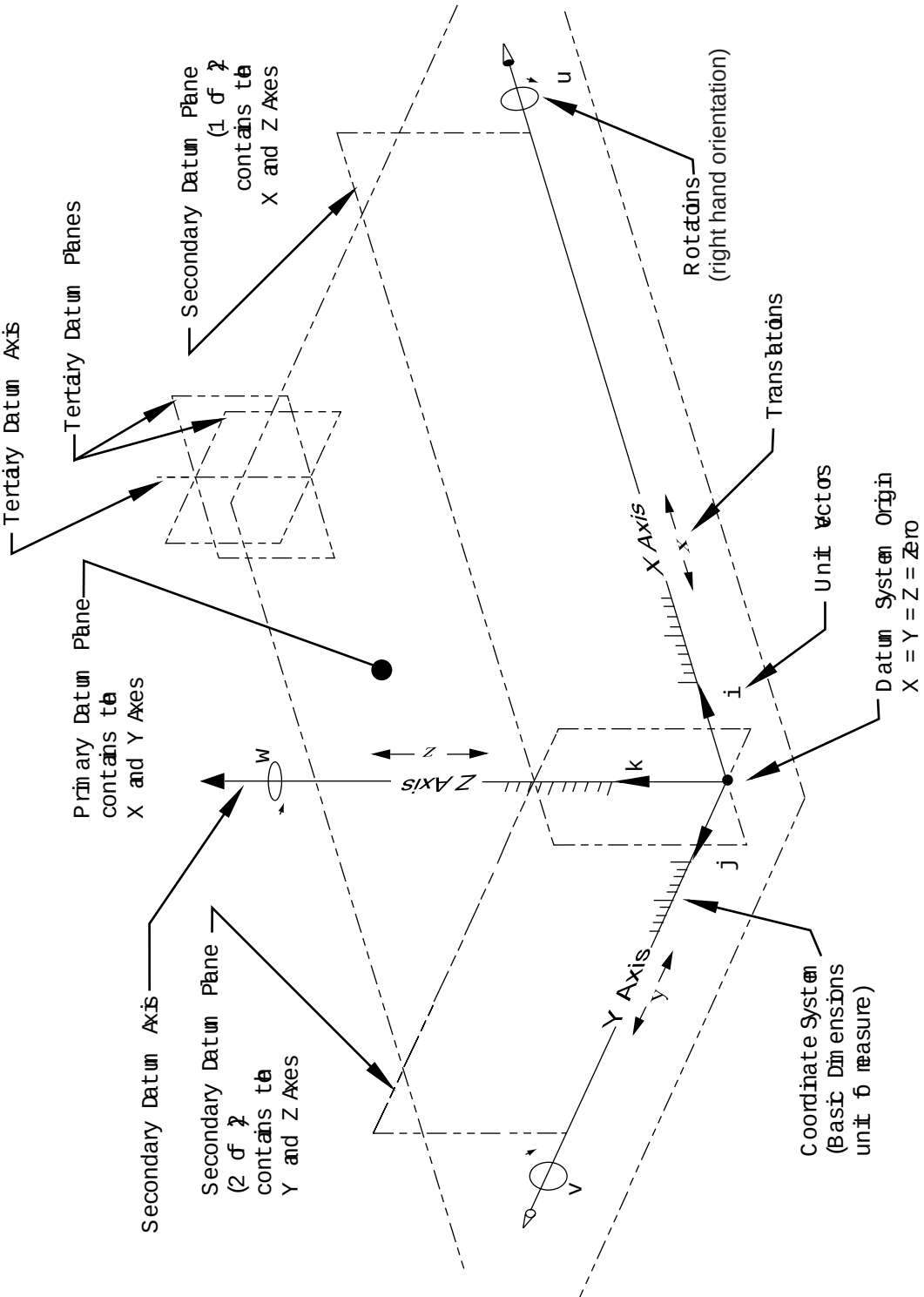


Figure 32 – Datum_system

4.2.236.1 datum_usages

The datum_usages is an inverse relationship that specifies that the existence of the Datum_system is dependent on the existence of the Datum_usage_in_datum_system_relationship (see 4.2.243) that specifies the Datum_system as its established_datum_system. There shall be one, two, or three Datum_usage_in_datum_system_relationship for the Datum_system.

4.2.236.2 defining_conditions

The defining_conditions is an inverse relationship that specifies that the existence of the Datum_system is dependent on the existence of the Datum_system_defining_conditions (see 4.2.237) that specifies the Datum_system as its conditions_of.

4.2.236.3 established_datum_reference_frame

The established_datum_reference_frame specifies the Datum_reference_frame (see 4.2.235) that is established by the Datum_system.

Constraints:

WR1: Any particular Datum_feature shall be used no more than one time to establish the Datum_system.

WR2: Any particular Datum_target shall be used no more than one time to establish the Datum_system.

NOTE EXPRESS specification:

```

*)
ENTITY datum_system
  SUBTYPE OF (non_feature_shape_element);
    established_datum_reference_frame : datum_reference_frame;
  INVERSE
    datum_usages : SET [1:3] OF
                  datum_usage_in_datum_system_relationship FOR
                  established_datum_system;
    defining_conditions : datum_system_defining_conditions FOR conditions_of;
END_ENTITY; -- datum_system

( *
```

4.2.237 Datum_system_defining_conditions

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_system_defining_conditions is a type of Managed_design_object (see 4.2.524). A Datum_system_defining_conditions is either a Datum_system_defining_conditions_without_material_conditions (see 4.2.239) or a Datum_system_defining_conditions_with_material_conditions (see 4.2.238). A Datum_system_defining_conditions is the specification of the conditions under which a Datum_system (see 4.2.236) is established. The conditions include the order in which the Datum (see 4.2.214) are established within the Datum_system and any material condition properties that are explicitly applied to Datum_feature (see 4.2.226) that participate in the establishment of the Datum_system.

The data associated with a Datum_system_defining_conditions are the following:

- assigned_datum_precedences;
- conditions_of.

4.2.237.1 assigned_datum_precedences

The assigned_datum_precedences specify the Datum_precedence_assignment (see 4.2.234) that define the order in which the Datum (see 4.2.214) are established in the Datum_system (see 4.2.236) specified as the conditions_of for the Datum_system_defining_conditions. There shall be one, two, or three Datum_precedence_assignment for the Datum_system_defining_conditions.

4.2.237.2 conditions_of

The conditions_of specifies the role of the Datum_system (see 4.2.236) for which the Datum_system_defining_conditions are defined.

Constraints:

WR1: The conditions_of the Datum_system_defining_conditions shall be the same instance of Datum_system as specified by the established_datum_system attribute of the Datum_usage_in_datum_system_-relationship that is specified by the assigned_to attribute of the Datum_precedence_assignment specified by the assigned_datum_precedences. That is, the Datum_system_defining_conditions and the specified Datum_precedence_assignments must be associated with the same Datum_system.

WR2: The name tertiary shall not exist in the names of the assigned_datum_precedences unless the name secondary exists in the names of the assigned_datum_precedences.

WR3: The name secondary shall not exist in the names of the assigned_datum_precedences unless the name primary exists in the names of the assigned_datum_precedences.

WR4: The datum_system_defining_conditions shall be referenced by at least one Geometric_tolerance_with_referenced_datum_system or Dimension_location_with_specified_datum_system.

NOTE EXPRESS specification:

```

*)
ENTITY datum_system_defining_conditions
  ABSTRACT SUPERTYPE OF (ONEOF (
    datum_system_defining_conditions_with_material_conditions,
    datum_system_defining_conditions_without_material_conditions))
  SUBTYPE OF (managed_design_object);
  conditions_of          : datum_system;
  assigned_datum_precedences : SET [1:3] OF datum_precedence_assignment;
END_ENTITY; -- datum_system_defining_conditions

( *

```

4.2.238 Datum_system_defining_conditions_with_material_conditions

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_system_defining_conditions_with_material_conditions is a type of Datum_system_defining_conditions (see 4.2.237) that specifies the conditions under which a Datum_system (see 4.2.236) is established and in which either the "least_material_requirement" or the "maximum_material_principle" has been applied to usages of one or more Datum_feature (see 4.2.226) in the Datum_system.

NOTE If neither the "least_material_requirement" nor the "maximum_material_principle" is explicitly applied and the Datum_feature is a feature of size, the principle that is termed "regardless of feature size" in ASME Y14.5M is assumed to be in effect.

The data associated with a Datum_system_defining_conditions_with_material_conditions are the following:

— applied_material_condition_properties.

4.2.238.1 applied_material_condition_properties

The applied_material_condition_properties specify the Datum_feature_material_condition_property (see 4.2.227) that define the material condition properties that are explicitly applied to the Datum_feature (see 4.2.226) in the context of the Datum_system (see 4.2.236) for the Datum_system_defining_conditions_with_material_conditions. There shall be one or more Datum_feature_material_condition_property for a Datum_system_defining_conditions_with_material_conditions.

EXAMPLE 1 The requirements may be such that the "maximum material principle" applies to Datum_feature A when it is used in one Datum_system. However, additional requirements may be such that neither the "maximum material principle" nor the "least material requirement" applies to Datum_feature A when it is used in a second Datum_system.

Constraints:

WR1: The conditions_of the Datum_system_defining_conditions_with_material_conditions shall be the same instance of Datum_system as specified by the established_datum_system attribute of the Datum_feature_usage_in_datum_system_relationship that is specified by the applied_to attribute of the Datum_feature_material_condition_property specified by the applied_material_condition_properties. That is, the Datum_system_defining_conditions_with_material_conditions and the specified Datum_feature_material_condition_property shall be associated with the same Datum_system.

NOTE EXPRESS specification:

```

*)
ENTITY datum_system_defining_conditions_with_material_conditions
  SUBTYPE OF (datum_system_defining_conditions);
  applied_material_condition_properties :
SET [1:?] OF datum_feature_material_condition_property;
END_ENTITY; -- datum_system_defining_conditions_with_material_conditions

( *
```

4.2.239 Datum_system_defining_conditions_without_material_conditions

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_system_defining_conditions_without_material_conditions is a type of Datum_system_defining_conditions (see 4.2.237) that specifies the conditions under which a Datum_system (see 4.2.236) is established and in which neither the "least material requirement" nor the "maximum material principle" have been applied to any of the Datum_feature (see 4.2.226) used to establish the Datum_system.

NOTE EXPRESS specification:

```

*)
ENTITY datum_system_defining_conditions_without_material_conditions
  SUBTYPE OF (datum_system_defining_conditions);
END_ENTITY; -- datum_system_defining_conditions_without_material_conditions

( *
```

4.2.240 Datum_target

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_target is a type of Physical_feature_or_part_template (see 4.2.658) that is used to define one or more Datum_target_set (see 4.2.241).

NOTE 1 Typically datum targets are used when it is not appropriate to specify an entire surface as a Datum_-feature (see 4.2.226).

NOTE 2 The definition of datum target is given in 3.4 of ISO 5459.

The data associated with a Datum_target are the following:

— datum_target_usages.

4.2.240.1 datum_target_usages

The datum_target_usages is an inverse relationship that specifies that the existence of the Datum_target is dependent on the existence of the Datum_target_usage_in_datum_target_set_relationship (see 4.2.242) that specifies the Datum_target as its used_datum_target. There shall be one or more Datum_target_usage_in_datum_target_set_relationship for the Datum_target.

NOTE EXPRESS specification:

```

*)
ENTITY datum_target
  SUBTYPE OF (physical_feature_or_part_template);
  INVERSE
    datum_target_usages : SET [1:?] OF
                        datum_target_usage_in_datum_target_set_relationship FOR
                        used_datum_target;
END_ENTITY; -- datum_target

( *
```

4.2.241 Datum_target_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_target_set is a type of Datum_feature (see 4.2.226) that is composed of one or more Datum_target (see 4.2.240).

The data associated with a Datum_target_set are the following:

— included_datum_targets.

4.2.241.1 included_datum_targets

The included_datum_targets is an inverse relationship that specifies that the existence of the Datum_target_set is dependent on the existence of the Datum_target_usage_in_datum_target_set_relationship (see 4.2.242) that specifies the Datum_target_set as its defined_datum_target_set. There shall be one or more Datum_target_usage_in_datum_target_set_relationship for the Datum_target_set.

Constraints:

WR1: A Datum_target_set shall not participate as a related_shape_element in a Shape_element_constituent_relationship. That is, a Datum_target_set shall not be a constituent of another Shape_element.

NOTE EXPRESS specification:

```

*)
ENTITY datum_target_set
  SUBTYPE OF (datum_feature);
  INVERSE
    included_datum_targets : SET [1:?] OF
      datum_target_usage_in_datum_target_set_relationship FOR
      defined_datum_target_set;
END_ENTITY; -- datum_target_set

( *
```

4.2.242 Datum_target_usage_in_datum_target_set_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

Datum_target_usage_in_datum_target_set_relationship is a type of Managed_design_object (see 4.2.524). A Datum_target_usage_in_datum_target_set_relationship is a relationship that associates a Datum_target (see 4.2.240) with a Datum_target_set (see 4.2.241) that the Datum_target participates in defining.

The data associated with a Datum_target_usage_in_datum_target_set_relationship are the following:

- datum_target_number;
- defined_datum_target_set;
- used_datum_target.

4.2.242.1 datum_target_number

The datum_target_number specifies the positive integer that identifies the usage of the Datum_target (see 4.2.240) in the definition of a Datum_target_set (see 4.2.241) for the Datum_target_usage_in_datum_target_set_relationship.

NOTE The datum target number is described in 7.1.1 of ISO 5459.

4.2.242.2 defined_datum_target_set

The defined_datum_target_set specifies the Datum_target_set (see 4.2.241) in which the specified Datum_target (see 4.2.240) participates in defining.

4.2.242.3 used_datum_target

The `used_datum_target` specifies the `Datum_target` (see 4.2.240) that is used to define the `Datum_target_set` (see 4.2.241) specified as the `defined_datum_target_set` for the `Datum_target_usage_in_datum_target_set_relationship`.

NOTE Since a `Datum_target` may be used to define multiple `Datum_target_sets` and the `Datum_target` number may be different depending on the `Datum_target_set` that the `Datum_target` participates in defining, the `Datum_target_number` is assigned to the `Datum_target_usage_in_datum_target_set_relationship` rather than on the `Datum_target`.

EXAMPLE 1 A datum target is used to define datum target sets A and G. The usage of the datum target in the definition of datum target set A is assigned a datum target number of 3. The usage of the datum target in defining datum target set G is assigned a datum target number of 1. This situation would be represented by a single instance of `Datum_target`, two instances of `Datum_target_set`, and two instances of `Datum_target_usage_in_datum_target_set_relationship`. The instances of `Datum_target_set` would be assigned the identifications of "A" and "G". The instances of `Datum_target_usage_in_datum_target_set_relationship` that associate the instance of `Datum_target` to the instances of `Datum_target_set` identified as "A" and "G" would have `datum_target_number` values of 3 and 1, respectively.

Constraints:

UR1: The combination of `used_datum_target` and `defined_datum_target_set` shall be unique within a population of `Datum_target_usage_in_datum_target_set_relationship`. That is, a specific `Datum_target` shall not be used more than once to define a specific `Datum_feature`.

UR2: The combination of `datum_target_number` and `defined_datum_target_set` shall be unique within a population of `Datum_target_usage_in_datum_target_set_relationship`.

NOTE EXPRESS specification:

```

*)
ENTITY datum_target_usage_in_datum_target_set_relationship
  SUBTYPE OF (managed_design_object);
    defined_datum_target_set : datum_target_set;
    used_datum_target        : datum_target;
    datum_target_number      : positive_integer;
  UNIQUE
    UR1 : used_datum_target, defined_datum_target_set;
    UR2 : datum_target_number, defined_datum_target_set;
END_ENTITY; -- datum_target_usage_in_datum_target_set_relationship

( *
```

4.2.243 Datum_usage_in_datum_system_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Datum_usage_in_datum_system_relationship is a type of Managed_design_object (see 4.2.524). A Datum_usage_in_datum_system_relationship is a relationship that associates a Datum_system (see 4.2.236) with a Datum (see 4.2.214) that participates in establishing the Datum_system.

The data associated with a Datum_usage_in_datum_system_relationship are the following:

- established_datum_system;
- precedence_assignment;
- used_datum.

4.2.243.1 established_datum_system

The established_datum_system specifies the Datum_system (see 4.2.236) that the Datum (see 4.2.214) specified as the used_datum participates in establishing for the Datum_usage_in_datum_system_relationship.

4.2.243.2 precedence_assignment

The precedence_assignment is an inverse relationship that specifies that the existence of the Datum_usage_in_datum_system_relationship is dependent on the existence of the Datum_precedence_assignment (see 4.2.234) that specifies the Datum_usage_in_datum_system_relationship as its assigned_to.

4.2.243.3 used_datum

The used_datum specifies the Datum (see 4.2.214) that is used to establish the Datum_system (see 4.2.236) specified as the established_datum_system for the Datum_usage_in_datum_system_relationship.

Constraints:

UR1: The combination of used_datum and established_datum_system shall be unique within a population of Datum_usage_in_datum_system_relationship. That is, a specific Datum shall not be used more than once to establish a specific Datum_system.

WR1: The Datum specified as the used_datum shall either be a Common_datum or a Single_datum.

IP1: The Datum specified as the used_datum shall either be a constituent Datum of the Datum_reference_frame established by the Datum_system specified by the established_datum_system, or it shall be dimensioned from one or more constituent Datum of the established Datum_reference_frame.

NOTE EXPRESS specification:

```

*)
ENTITY datum_usage_in_datum_system_relationship
  SUBTYPE OF (managed_design_object);
    established_datum_system : datum_system;
    used_datum                : datum;
  INVERSE
    precedence_assignment : datum_precedence_assignment FOR assigned_to;
  UNIQUE
    UR1 : used_datum, established_datum_system;
END_ENTITY; -- datum_usage_in_datum_system_relationship

```

(*

4.2.244 Default_assembly_bond_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_assembly_bond_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Default_assembly_bond_shape is the shape of an Assembly_bond_definition (see 4.2.33).

The data associated with a Default_assembly_bond_shape are the following:

— characterized_bond_definition.

4.2.244.1 characterized_bond_definition

The characterized_bond_definition specifies the role of the Assembly_bond_definition (see 4.2.33) for the Default_assembly_bond_shape.

NOTE EXPRESS specification:

```

*)
ENTITY default_assembly_bond_shape
  SUBTYPE OF (bound_volume_shape, feature_shape_definition);
    characterized_bond_definition : assembly_bond_definition;
END_ENTITY; -- default_assembly_bond_shape

```

(*

4.2.245 Default_attachment_size_and_component_termination_passage_based_land_physical_template

A Default_attachment_size_and_component_termination_passage_based_land_physical_template is a type of Default_attachment_size_based_land_physical_template (see 4.2.247) and of Default_component_termination_passage_based_land_physical_template (see 4.2.248). The shape of this template is dependent on both the requirements from the passage definition and the requirement for support of the interface between the instance of the Part_terminal (see 4.2.630) intended to mate with an instance of the Land_physical_template (see 4.2.493).

NOTE EXPRESS specification:

```
*)
ENTITY default_attachment_size_and_component_termination_passage_based_land_physical_template
  SUBTYPE OF (
    default_component_termination_passage_based_land_physical_template,
    default_attachment_size_based_land_physical_template);
END_ENTITY;
-- default_attachment_size_and_component_termination_passage_based_land_physical_template
(*
```

4.2.246 Default_attachment_size_and_via_based_land_physical_template

A Default_attachment_size_and_via_based_land_physical_template is a type of Default_attachment_size_based_land_physical_template (see 4.2.247) and of Default_via_based_land_physical_template (see 4.2.255). This template shape is dependent on both the requirement from the Default_via_definition (see 4.2.256) and the shape of the related Part_terminal (see 4.2.630).

NOTE EXPRESS specification:

```
*)
ENTITY default_attachment_size_and_via_based_land_physical_template
  SUBTYPE OF (default_via_based_land_physical_template,
    default_attachment_size_based_land_physical_template);
END_ENTITY; -- default_attachment_size_and_via_based_land_physical_template
(*
```

4.2.247 Default_attachment_size_based_land_physical_template

A Default_attachment_size_based_land_physical_template is a type of Default_land_physical_template (see 4.2.250). A Default_attachment_size_based_land_physical_template may be a Default_attachment_size_and_component_termination_passage_based_land_physical_template (see 4.2.245) or a Default_attachment_size_and_via_based_land_physical_template (see 4.2.246). This template shape is dependent only on the shape of the Part_terminal (see 4.2.630) it is intended to mate with.

The data associated with a Default_attachment_size_based_land_physical_template are the following:

- interface_access_mechanisms;
- minimum_attachment_region_size;
- maximum_attachment_region_size.

4.2.247.1 interface_access_mechanisms

The interface_access_mechanisms is an inverse relationship that specifies that the existence of the Default_attachment_size_based_land_physical_template is dependent on the existence of the Land_template_interface_terminal (see 4.2.494) that specifies the Default_attachment_size_based_land_physical_template as its associated_definition. There shall be one or more Land_template_interface_terminal for a Default_attachment_size_based_land_physical_template. Fabrication and assembly tolerances define the required shapes and variances for the terminal shapes.

4.2.247.2 minimum_attachment_region_size

The minimum_attachment_region_size specifies one of the roles of the Connection_zone for the Default_attachment_size_based_land_physical_template. The minimum_attachment_region_size specifies the smallest Connection_zone for the Default_attachment_size_based_land_physical_template.

4.2.247.3 maximum_attachment_region_size

The maximum_attachment_region_size specifies one of the roles of the Connection_zone for the Default_attachment_size_based_land_physical_template. The maximum_attachment_region_size specifies the largest Connection_zone for the Default_attachment_size_based_land_physical_template.

NOTE EXPRESS specification:

```

*)
ENTITY default_attachment_size_based_land_physical_template
  SUPERTYPE OF (ONEOF (
    default_attachment_size_and_component_termination_passage_based_land_physical_template,
    default_attachment_size_and_via_based_land_physical_template))
  SUBTYPE OF (default_land_physical_template);
  minimum_attachment_region_size : OPTIONAL connection_zone;
  maximum_attachment_region_size : OPTIONAL connection_zone;
  INVERSE
    interface_access_mechanisms : SET [1:?] OF land_template_interface_terminal FOR
      associated_definition;
WHERE
  WR1: NOT(EXISTS(minimum_attachment_region_size) AND
    EXISTS(maximum_attachment_region_size)) OR
    (minimum_attachment_region_size :&#60;&#62;;
    maximum_attachment_region_size);
END_ENTITY; -- default_attachment_size_based_land_physical_template
( *

```

4.2.248 Default_component_termination_passage_based_land_physical_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_component_termination_passage_based_land_physical_template is a type of Default_passage_based_land_physical_template (see 4.2.251). A Default_component_termination_passage_based_land_physical_template may be a Default_attachment_size_and_component_termination_passage_based_land_physical_template (see 4.2.245). The shape of the land template is dependent upon the requirements from Passage_technology (see 4.2.638).

The data associated with a Default_component_termination_passage_based_land_physical_template are the following:

— of_passage_technology.

4.2.248.1 of_passage_technology

The of_passage_technology specifies the role of the Default_component_termination_passage_definition (see 4.2.249) for the Default_component_termination_passage_based_land_physical_template.

NOTE EXPRESS specification:

```

*)
ENTITY default_component_termination_passage_based_land_physical_template
  SUPERTYPE OF
    (default_attachment_size_and_component_termination_passage_based_land_physical_template)
  SUBTYPE OF (default_passage_based_land_physical_template);
  SELF\default_passage_based_land_physical_template.of_passage_technology :
    default_component_termination_passage_definition;
END_ENTITY; -- default_component_termination_passage_based_land_physical_template

( *

```

4.2.249 Default_component_termination_passage_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_component_termination_passage_definition is a type of Passage_technology (see 4.2.638) that defines a default passage based on characteristics of the associated Part_terminal (see 4.2.630).

The data associated with a Default_component_termination_passage_definition are the following:

- deposition_material;
- maximum_allowed_component_terminal_extent;
- minimum_allowed_component_terminal_extent;
- minimum_as_finished_deposition_thickness;
- minimum_as_finished_passage_extent.

4.2.249.1 deposition_material

The deposition_material specifies the role of the Ee_material (see 4.2.318) for the Default_component_termination_passage_definition.

4.2.249.2 maximum_allowed_component_terminal_extent

The maximum_allowed_component_terminal_extent specifies one role of the Length_data_element (see 4.2.511) for the Default_component_termination_passage_definition.

4.2.249.3 minimum_allowed_component_terminal_extent

The minimum_allowed_component_terminal_extent specifies one role of the Length_data_element (see 4.2.511) for the Default_component_termination_passage_definition.

4.2.249.4 minimum_as_finished_deposition_thickness

The minimum_as_finished_deposition_thickness specifies one role of the Length_data_element (see 4.2.511) for the Default_component_termination_passage_definition.

4.2.249.5 minimum_as_finished_passage_extent

The minimum_as_finished_passage_extent specifies one role of the Length_data_element (see 4.2.511) for the Default_component_termination_passage_definition.

Constraints:

WR1: The passage deposition material shall be conductive.

NOTE EXPRESS specification:

*)

ENTITY default_component_termination_passage_definition

SUBTYPE OF (passage_technology);

minimum_allowed_component_terminal_extent

: length_data_element

maximum_allowed_component_terminal_extent

: length_data_element

SELF\passage_technology.deposition_material

: ee_material;

SELF\passage_technology.minimum_as_finished_deposition_thickness

: length_data_element

SELF\passage_technology.minimum_as_finished_passage_extent

: length_data_element

END_ENTITY; -- default_component_termination_passage_definition

(*

4.2.250 Default_land_physical_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_land_physical_template is a type of Land_physical_template (see 4.2.493). A Default_land_physical_template is either a Default_passage_based_land_physical_template (see 4.2.251) or a Default_attachment_size_based_land_physical_template (see 4.2.247).

The data associated with a Default_land_physical_template are the following:

— of_stratum_technology.

4.2.250.1 of_stratum_technology

The of_stratum_technology specifies the role of the Design_technology (see ??) for the Default_land_physical_template.

NOTE EXPRESS specification:

```

*)
ENTITY default_land_physical_template
  ABSTRACT SUPERTYPE OF (ONEOF (
    default_passage_based_land_physical_template,
    default_attachment_size_based_land_physical_template))
  SUBTYPE OF (land_physical_template);
  SELF\stratum_feature_template.of_stratum_technology :
    design_layer_technology;
END_ENTITY; -- default_land_physical_template

( *

```

4.2.251 Default_passage_based_land_physical_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_passage_based_land_physical_template is a type of Default_land_physical_template (see 4.2.250). A Default_passage_based_land_physical_template is either a Default_component_termination_passage_based_land_physical_template (see 4.2.248), a Default_unsupported_passage_based_land_physical_template (see 4.2.253), or a Default_via_based_land_physical_template (see 4.2.255). The shape of a Land (see 4.2.490) based on this template is dependent on the shape of the passage with which it is associated.

EXAMPLE 1 Figure 33 illustrates the relationship between the minimum_annular_ring and the size category of finished hole for which the minimum_annular_ring is defined. The category information is specified in the member of Passage_technology (see 4.2.638) associated with this Default_passage_based_land_physical_template.

NOTE The usual domain technical reference is ANSI/IPC D-275.

The data associated with a Default_passage_based_land_physical_template are the following:

- minimum_annular_ring;
- minimum_fabrication_allowance;
- of_passage_technology;
- of_stratum_extent.

4.2.251.1 minimum_annular_ring

The minimum_annular_ring specifies the Length_data_element (see 4.2.511) of the smallest dimension between each point on the inner bound of a passage and the nearest point on the outer bound of a Land (see 4.2.490) containing the passage for the Default_passage_based_land_physical_template.

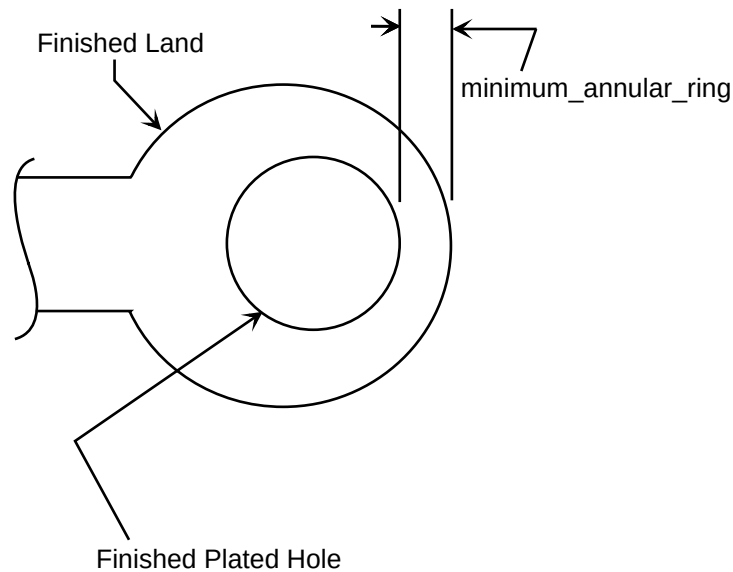


Figure 33 – Measurement arrangement for minimum_annular_ring

4.2.251.2 minimum_fabrication_allowance

The minimum_fabrication_allowance specifies the Length_data_element (see 4.2.511) for the smallest dimension of the total of the manufacturing process tolerance effects for the Default_passage_based_land_physical_template.

4.2.251.3 of_passage_technology

The of_passage_technology specifies the role of the Passage_technology (see 4.2.638) for the Default_passage_based_land_physical_template.

4.2.251.4 of_stratum_extent

The of_stratum_extent specifies the role of the Inter_stratum_extent (see 4.2.431) for the Default_passage_based_land_physical_template. The of_stratum_extent specifies the actual subset of the passage specified by of_passage_technology to which this Default_passage_based_land_physical_template applies.

Constraints:

UR1: The combination of the attribute of_passage_technology and the attribute of_stratum_technology inherited from the supertype Default_land_physical_template shall be unique in a population of Default_passage_based_land_physical_template.

WR1: The stratum extent specified by of_stratum_extent shall be a subset of the stratum extent specified by the of_passage_technology.

NOTE EXPRESS specification:

*)

```
ENTITY default_passage_based_land_physical_template
  ABSTRACT SUPERTYPE OF (ONEOF (default_via_based_land_physical_template,
    default_component_termination_passage_based_land_physical_template,
    default_unsupported_passage_based_land_physical_template))
  SUBTYPE OF (default_land_physical_template);
  minimum_fabrication_allowance : length_data_element;
  of_passage_technology          : passage_technology;
  of_stratum_extent              : inter_stratum_extent;
  minimum_annular_ring          : length_data_element;
  UNIQUE
  UR1 : of_passage_technology, SELF\default_land_physical_template.of_stratum_technology
END_ENTITY; -- default_passage_based_land_physical_template
```

(*

4.2.252 Default_trace_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_trace_template is a type of Trace_template (see 4.2.930). The Default_trace_template is associated with a particular Stratum_technology (see 4.2.871), to allow verification that the design of the trace template meets the relevant minimum and maximum guidelines of that Stratum_technology.

The data associated with a Default_trace_template are the following:

— of_stratum_technology.

4.2.252.1 of_stratum_technology

The of_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Default_trace_template.

NOTE EXPRESS specification:

*)

```
ENTITY default_trace_template
  SUBTYPE OF (trace_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- default_trace_template
```

(*

4.2.253 Default_unsupported_passage_based_land_physical_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_unsupported_passage_based_land_physical_template is a type of Default_passage_based_land_physical_template (see 4.2.251) where the passage type is a passage that has no metallized support.

The data associated with a Default_unsupported_passage_based_land_physical_template are the following:

— of_passage_technology.

4.2.253.1 of_passage_technology

The of_passage_technology specifies the role of the Default_unsupported_passage_definition (see 4.2.254) for the Default_unsupported_passage_based_land_physical_template.

NOTE EXPRESS specification:

```
*)
ENTITY default_unsupported_passage_based_land_physical_template
  SUBTYPE OF (default_passage_based_land_physical_template);
  SELF\default_passage_based_land_physical_template.of_passage_technology :
  default_unsupported_passage_definition;
END_ENTITY; -- default_unsupported_passage_based_land_physical_template

( *
```

4.2.254 Default_unsupported_passage_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_unsupported_passage_definition is a type of Passage_technology (see 4.2.638) where the passage is unsupported by metallization.

The data associated with a Default_unsupported_passage_definition are the following:

— minimum_as_finished_passage_definition.

4.2.254.1 minimum_as_finished_passage_definition

The minimum_as_finished_passage_definition specifies one role of the Length_data_element (see 4.2.511) for the Default_via_definition (see 4.2.256).

NOTE EXPRESS specification:

```

*)
ENTITY default_unsupported_passage_definition
  SUBTYPE OF (passage_technology);
  SELF\passage_technology.minimum_as_finished_passage_extent : length_data_element;
END_ENTITY; -- default_unsupported_passage_definition

( *

```

4.2.255 Default_via_based_land_physical_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_via_based_land_physical_template is a type of Default_passage_based_land_physical_template (see 4.2.251) where the passage type is a Via (see 4.2.947). A Default_via_based_land_physical_template may be a Default_attachment_size_and_via_based_land_physical_template (see 4.2.246).

The data associated with a Default_via_based_land_physical_template are the following:

— of_passage_technology.

4.2.255.1 of_passage_technology

The of_passage_technology specifies the role of the Default_via_definition (see 4.2.256) for the Default_via_based_land_physical_template.

NOTE EXPRESS specification:

```

*)
ENTITY default_via_based_land_physical_template
  SUPERTYPE OF (
    default_attachment_size_and_via_based_land_physical_template)
  SUBTYPE OF (default_passage_based_land_physical_template);
  SELF\default_passage_based_land_physical_template.of_passage_technology :
  default_via_definition;
END_ENTITY; -- default_via_based_land_physical_template

( *

```

4.2.256 Default_via_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Default_via_definition is a type of Passage_technology (see 4.2.638) that defines the default shape of the Via (see 4.2.947).

The data associated with a Default_via_definition are the following:

- deposition_material;
- minimum_as_finished_deposition_thickness;
- minimum_as_finished_passage_extent.

4.2.256.1 deposition_material

The deposition_material specifies the role of the Ee_material (see 4.2.318) in the Via (see 4.2.947) for the Default_via_definition.

4.2.256.2 minimum_as_finished_deposition_thickness

The minimum_as_finished_deposition_thickness specifies the Length_data_element (see 4.2.511) that is the lower bound for the thickness of the deposited Ee_material (see 4.2.318) for the Default_via_definition.

4.2.256.3 minimum_as_finished_passage_extent

The minimum_as_finished_passage_extent specifies one role of the Length_data_element (see 4.2.511) for the Default_via_definition.

Constraints:

WR1: The passage deposition material shall be conductive.

NOTE EXPRESS specification:

*)

ENTITY default_via_definition

 SUBTYPE OF (passage_technology);

 SELF\passage_technology.deposition_material : ee_material;

 SELF\passage_technology.minimum_as_finished_deposition_thickness : length_data_element

 SELF\passage_technology.minimum_as_finished_passage_extent : length_data_element

END_ENTITY; -- default_via_definition

(*

4.2.257 Delete_design_object_management_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Delete_design_object_management_relationship is a type of Design_object_management_relationship (see 4.2.278). Together with Change_design_object_management_relationship and Add_design_object_management_relationship, Delete_design_object_management_relationship supports the exchange of changes to a design by the use of a data set containing only (1) the information necessary to identify the design (e.g., an Ee_product_definition or an Ee_product_version), which is assumed to be already known to the receiving system, and (2) the design objects that have been added, changed, or deleted.

NOTE Note that all the quality and structural requirements defined in this part of ISO 10303 also apply to the resultant new design iteration.

The data associated with a Delete_design_object_management_relationship are the following:

— previous_design_object.

4.2.257.1 previous_design_object

The previous_design_object specifies the role of the Managed_design_object (see 4.2.524) for the Delete_design_object_management_relationship.

Constraints:

UR1: The combination of the previous_design_object and the previous_design inherited from Design_object_management_relationship shall be unique within a population of Delete_design_object_management_relationships.

NOTE EXPRESS specification:

```
*)
ENTITY delete_design_object_management_relationship
  SUBTYPE OF (design_object_management_relationship);
  previous_design_object : managed_design_object;
  UNIQUE
    UR1 : previous_design_object, previous_design;
END_ENTITY; -- delete_design_object_management_relationship

( *
```

4.2.258 Dependent_material_removal_feature_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dependent_material_removal_feature_template is a type of Material_removal_feature_template (see 4.2.534).

NOTE EXPRESS specification:

```
* )
ENTITY dependent_material_removal_feature_template
  SUBTYPE OF (material_removal_feature_template);
  associated_template : default_land_physical_template;
END_ENTITY; -- dependent_material_removal_feature_template

--tt tc1 wd9 modified
(*
```

4.2.259 **Dependently_located_layer_connection_point**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dependently_located_layer_connection_point is a type of Layer_connection_point (see 4.2.499) that is position dependent on the location of an Interconnect_module_terminal (see 4.2.463) or an Inter_stratum_feature (see 4.2.432).

NOTE This part of ISO 10303 provides the capability to support one interconnect design for multiple assembly designs where components at the assembly level may or may not exist for any or all of the multiple assembly designs.

EXAMPLE 1 An example of an Dependently_located_layer_connection_point is the case where a common interconnect must be supplied for several assemblies. A specific microcircuit device, such as "U32" may be supplied for one or more, but not all, of the assemblies, but the interconnect must be identical.

EXAMPLE 2 Figure 34 illustrates a detail of a layout drawing that shows the relationship of two members of Dependently_located_layer_connection_point and the associated Conductor (see 4.2.176).

The data associated with a Dependently_located_layer_connection_point are the following:

— associated_design_object.

4.2.259.1 **associated_design_object**

The associated_design_object specifies either the role of an Interconnect_module_terminal (see 4.2.463) or the role of an Inter_stratum_feature (see 4.2.432) for the Dependently_located_layer_connection_point.

NOTE The associated_design_object may not fully constrain the location of the Dependently_located_layer_connection_point.

Constraints:

IP1: The Layer_connection_point is constrained to be within the material volume specified by the associated_design_object.

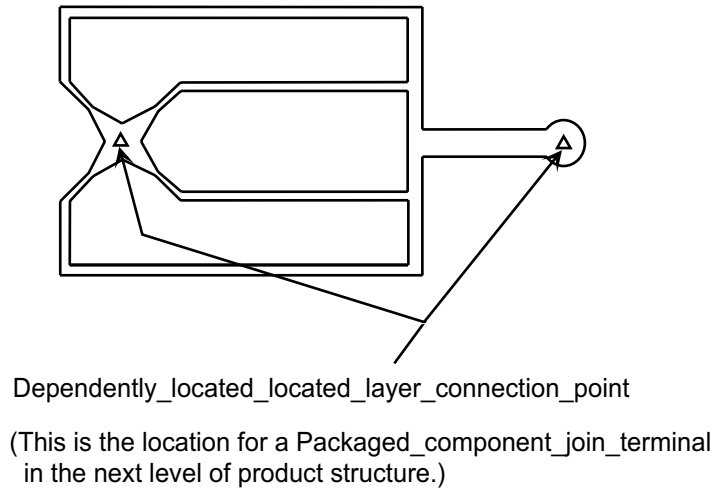


Figure 34 – Dependently_located_layer_connection_point

NOTE EXPRESS specification:

```

*)
ENTITY dependently_located_layer_connection_point
  SUBTYPE OF (layer_connection_point);
    associated_design_object : terminal_or_inter_stratum_feature;
    reference_zone : OPTIONAL connection_zone;
END_ENTITY; -- dependently_located_layer_connection_point
( *

```

4.2.260 Derived_shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Derived_shape_element is a type of Non_feature_shape_element (see 4.2.560) that is derived from a set of one or more other Shape_element (see 4.2.810). A Derived_shape_element is one of an Apex (see 4.2.27), a Centre_of_symmetry (see 4.2.91), a Geometric_alignment (see 4.2.416), a Geometric_intersection (see 4.2.417), a Linear_extension (see 4.2.518), or a Tangent_shape_element (see 4.2.885).

The data associated with a Derived_shape_element are the following:

— deriving_relationships.

4.2.260.1 deriving_relationships

The `deriving_relationships` is an inverse relationship that specifies that the existence of the `Derived_shape_element` is dependent on the existence of the `Shape_element_deriving_relationship` (see 4.2.814) that specifies the `Derived_shape_element` as its `relating_shape_element`. There shall be one or more `Shape_element_deriving_relationship` specified as the `deriving_relationships` for a `Derived_shape_element`.

EXAMPLE 1 The `Centre_axis` (see 4.2.90) of a cylindrical hole is a `Derived_shape_element`. The `Centre_axis` cannot exist unless the hole exists.

Constraints:

WR1: A `Derived_shape_element` shall not be derived from itself, either directly or through intermediate `Derived_shape_elements`.

NOTE EXPRESS specification:

```
*)
ENTITY derived_shape_element
  ABSTRACT SUPERTYPE OF (ONEOF (apex,centre_of_symmetry,linear_extension,
    geometric_intersection,geometric_alignment,tangent_shape_element))
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    deriving_relationships : SET [1:?] OF
      shape_element_deriving_relationship FOR
      relating_shape_element;
END_ENTITY; -- derived_shape_element

( *
```

4.2.261 Design_characteristic_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Design_characteristic_occurrence` is a type of `Ee_requirement_occurrence` (see 4.2.330) that specifies the technical performance values and tolerances for some aspect of a design.

NOTE The specifications set forth in `Design_characteristics` may include such items as enterprise or customer specifications on preferred packaging, power design and control, design rules, and preferred design practices.

NOTE EXPRESS specification:

```

*)
ENTITY design_characteristic_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- design_characteristic_occurrence

( *
```

4.2.262 Design_composition_path

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_composition_path is a type of Managed_design_object (see 4.2.524). A Design_composition_path is a collection of Functional_unit (see 4.2.400) that together compose one path through the complete functional definition that has the design_functional_unit as a reference to the Functional_unit_definition (see 4.2.401) that is the leaf node. The functional definition has a root node that is a Functional_unit_network_definition (see 4.2.402). The path is then established by starting at the leaf element and traversing the collection of path_element in the sequence in which they exist in the functional definition between the leaf element and the root element. Each member of path_element establishes a link in the sequence. The Design_composition_path is a mapping source for a packaging application to use to map between a hierarchical description and a flattened description of a network.

NOTE The types of node elements that are composing a directed acyclic graph must be consistent. This part of ISO 10303 uses Functional_unit_definition (see 4.2.401) as the node elements.

The data associated with a Design_composition_path are the following:

- design_functional_unit;
- path_element.

4.2.262.1 design_functional_unit

The design_functional_unit specifies the Functional_unit (see 4.2.400) for the leaf element for the Design_composition_path.

4.2.262.2 path_element

The path_element specifies the role of the Functional_unit (see 4.2.400) for the Design_composition_path. There shall be one or more Functional_unit for a Design_composition_path.

NOTE EXPRESS specification:

```

*)
ENTITY design_composition_path
  SUBTYPE OF (managed_design_object);
  path_element      : SET [1:?] OF functional_unit;
  design_functional_unit : functional_unit;
END_ENTITY; -- design_composition_path

( *

```

4.2.263 Design_functional_unit_allocation_to_assembly_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_functional_unit_allocation_to_assembly_component is a type of Managed_design_object (see 4.2.524). A Design_functional_unit_allocation_to_assembly_component may be a Transmission_line_functional_unit_to_routed_printed_component_allocation (see 4.2.932). A Design_functional_unit_allocation_to_assembly_component is the relationship between the Design_composition_path (see 4.2.262) and the Assembly_component (see 4.2.34) that performs the required function.

The data associated with a Design_functional_unit_allocation_to_assembly_component are the following:

- functional_design_definition_path;
- global_swappable;
- implementation;
- local_swappable;
- of_property;
- path_alias;
- swap_code.

NOTE Frozen_assignment (see 4.2.394) is used to indicate that the Design_functional_unit_allocation_to_assembly_component relationship is approved and can not be changed during further design activity without additional formal action.

4.2.263.1 functional_design_definition_path

The functional_design_definition_path specifies the role of the Design_composition_path (see 4.2.262) for the Design_functional_unit_allocation_to_assembly_component.

4.2.263.2 **global_swappable**

The `global_swappable` specifies a boolean value that indicates if occurrences of the associated path can be exchanged within the context of an `Ee_product` being designed.

4.2.263.3 **implementation**

The `implementation` specifies the role of the `Assembly_component` (see 4.2.34) for the `Design_functional_unit_allocation_to_assembly_component`.

NOTE A `Design_functional_unit_allocation_to_assembly_component` approval may be indicated by including a complex instance consisting of `Design_functional_unit_allocation_to_assembly_component` and `Frozen_assignment` (see 4.2.394).

4.2.263.4 **local_swappable**

The `local_swappable` specifies a boolean value that indicates if occurrences of the associated path can be exchanged within the context of an `Assembly_component`.

4.2.263.5 **of_property**

The `of_property` specifies the role of the `Coordinated_characteristic` for the `Design_functional_unit_allocation_to_assembly_component`. There may be more than one `of_property` for a `Design_functional_unit_allocation_to_assembly_component`. The `of_property` need not be specified for a particular `Design_functional_unit_allocation_to_assembly_component`.

4.2.263.6 **path_alias**

The `path_alias` specifies the string that is the identifier in the context of the implementation for the `Design_functional_unit_allocation_to_assembly_component`.

EXAMPLE 1 "A", "B", "C", "D", are typical `path_alias` for a quad NAND gate part.

4.2.263.7 **swap_code**

The `swap_code` specifies the string that is the type of replacement that is valid.

Constraints:

UR1: The combination of `implementation` and `path_alias` shall be unique within a population of `Design_functional_unit_allocation_to_assembly_component`.

NOTE EXPRESS specification:

```

*)
ENTITY design_functional_unit_allocation_to_assembly_component
  SUPERTYPE OF (transmission_line_functional_unit_to_routed_printed_component_allocation)
  SUBTYPE OF (managed_design_object);
  functional_design_definition_path : design_composition_path;
  implementation                     : assembly_component;
  global_swappable                   : BOOLEAN;
  local_swappable                    : BOOLEAN;
  of_property                        : OPTIONAL SET [1:?] OF coordinated_characteristic;
  path_alias                         : ee_name;
  swap_code                          : STRING;
  UNIQUE
    UR1 : implementation, path_alias;
END_ENTITY; -- design_functional_unit_allocation_to_assembly_component

( *

```

4.2.264 Design_functional_unit_allocation_to_assembly_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_functional_unit_allocation_to_assembly_joint is a type of Managed_design_object (see 4.2.524).
 A Design_functional_unit_allocation_to_assembly_joint is an assignment of a Functional_unit (see 4.2.400)
 in a Design_composition_path (see 4.2.262) to its implementation in an Assembly_joint (see 4.2.41).

The data associated with a Design_functional_unit_allocation_to_assembly_joint are the following:

- functional_design_definition_path;
- implementation.

4.2.264.1 functional_design_definition_path

The functional_design_definition_path specifies the role of the Design_composition_path (see 4.2.262) for the Design_functional_unit_allocation_to_assembly_joint.

4.2.264.2 implementation

The implementation specifies the role of the Assembly_joint (see 4.2.41) for the Design_functional_unit_allocation_to_assembly_joint.

NOTE This application object is included to support detailed analysis of product design.

Constraints:

UR1: The functional_design_definition_path shall be unique within a population of Design_functional_unit_allocation_to_assembly_joint.

NOTE EXPRESS specification:

```

*)
ENTITY design_functional_unit_allocation_to_assembly_joint
  SUBTYPE OF (managed_design_object);
    functional_design_definition_path : design_composition_path;
    implementation                     : assembly_joint;
UNIQUE
  UR1: functional_design_definition_path;
END_ENTITY; -- design_functional_unit_allocation_to_assembly_joint

( *
```

4.2.265 Design_functional_unit_allocation_to_fabrication_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_functional_unit_allocation_to_fabrication_joint is a type of Managed_design_object (see 4.2.524).
 A Design_functional_unit_allocation_to_fabrication_joint is an assignment of a Functional_unit (see 4.2.400) in a Design_composition_path (see 4.2.262) to its implementation in a Fabrication_joint (see 4.2.368).

The data associated with a Design_functional_unit_allocation_to_fabrication_joint are the following:

- functional_design_definition_path;
- implementation.

4.2.265.1 functional_design_definition_path

The functional_design_definition_path specifies the role of the Design_composition_path (see 4.2.262) for the Design_functional_unit_allocation_to_fabrication_joint.

4.2.265.2 implementation

The implementation specifies the role of the Fabrication_joint (see 4.2.368) for the Design_functional_unit_allocation_to_fabrication_joint.

NOTE This application object is included to support detailed analysis of product design.

Constraints:

UR1: The functional_design_definition_path shall be unique within a population of Design_functional_unit_allocation_to_fabrication_joint.

NOTE EXPRESS specification:

```

*)
ENTITY design_functional_unit_allocation_to_fabrication_joint
  SUBTYPE OF (managed_design_object);
    functional_design_definition_path : design_composition_path;
    implementation                     : fabrication_joint;
UNIQUE
  UR1: functional_design_definition_path;
END_ENTITY; -- design_functional_unit_allocation_to_fabrication_joint

( *
```

4.2.266 Design_functional_unit_allocation_to_reference_functional_unit

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_functional_unit_allocation_to_reference_functional_unit is a type of Managed_design_object (see 4.2.524). A Design_functional_unit_allocation_to_reference_functional_unit is an assignment of a Functional_unit (see 4.2.400) in a Design_composition_path (see 4.2.262) to a Functional_unit in a Reference_composition_path (see 4.2.764) for the functional definition of a product.

NOTE 1 This provides the capability to exchange a functional decomposition that an enterprise reuses in multiple designs.

NOTE 2 The library functional hierarchy, if provided, is required to be a network definition with the constraint that only external references on the root node of the network structure will be mapped to physical terminals of the reused product design.

NOTE 3 The allocation relationship in this Application Object is critical in mapping between a functional design and a physical implementation of that design, because this part of ISO 10303 relies on the reference mapping between a functional definition in a library and the physical implementation of that functional definition in that library, usually in a packaged part, to identify the necessary definitions that need to be instantiated in the physical design.

The data associated with a Design_functional_unit_allocation_to_reference_functional_unit are the following:

- design_definition_path;
- reference_definition_path.

4.2.266.1 design_definition_path

The design_definition_path specifies the role of the Design_composition_path (see 4.2.262) for the Design_functional_unit_allocation_to_reference_functional_unit.

4.2.266.2 reference_definition_path

The reference_definition_path specifies the role of the Reference_composition_path (see 4.2.764) for the Design_functional_unit_allocation_to_reference_functional_unit.

Constraints:

WR1: The functional_unit_definition that defines the leaf node of the reference_definition_path and the functional_unit_definition that defines the leaf node of the design_definition_path shall be the same definition.

NOTE EXPRESS specification:

```
*)
ENTITY design_functional_unit_allocation_to_reference_functional_unit
  SUBTYPE OF (managed_design_object);
    reference_definition_path : reference_composition_path;
    design_definition_path   : design_composition_path;
END_ENTITY; -- design_functional_unit_allocation_to_reference_functional_unit

( *
```

4.2.267 Design_intent_modification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification is a type of Managed_design_object (see 4.2.524). A Design_intent_modification may be one of Material_addition_feature (see 4.2.528), Material_removal_feature (see 4.2.533), or Material_shape_modification (see 4.2.536). A Design_intent_modification provides capabilities to archive design intent along with constraint based modifications so that if a constraint is removed the original design intent, or a part thereof, may be restored.

EXAMPLE 1 This example shows how this part of ISO 10303 may be used to exchange or archive design intent for the case of a power plane where there is penetration of the power plane by other signals. Figure 35 is the original design intent for Power Plane 1 that is captured in a member of Connected_filled_area (see 4.2.184)(#1).

Figure 36 is the original design intent for Power Plane 2 that is captured in a member of Connected_filled_area (#2).

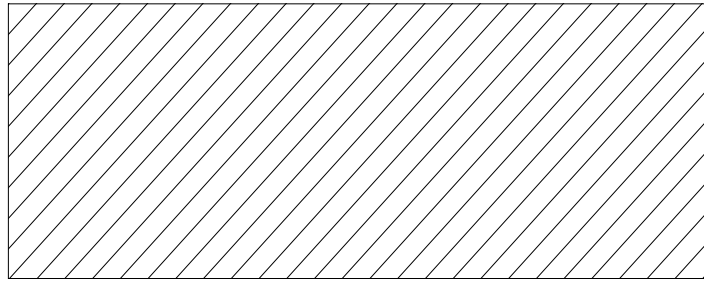


Figure 35 – Power Plane 1 design intent

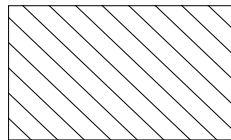


Figure 36 – Power Plane 2 design intent

Figure 37 is the original design intent for Signal (see 4.2.823) 3 that is captured in a member of Conductor (see 4.2.176)(#3). In order to accomplish the embedding of Power Plane 2 into Power Plane 1 and in order to accomplish the embedding of Signal 3 into the result of the previous embedding, a sequence is followed. In order to meet requirements, constraints must be satisfied regarding spacing of conductive areas. These constraints are specified by the `Ee_requirement_occurrence` (see 4.2.330). Note that the following method is based on the relative priority of the signal and the two power planes. This example assumes that the conductor has the highest priority for area, followed by Power Plane 2 followed by Power Plane 1 which has the lowest priority.



Figure 37 – Signal 3 design intent

Step 1: Create three "design intent" `Stratum_features` as above. Note that these features are a permanent part of the design.

Step 2: Create a member of `Material_removal_stratum_feature_template` (see figure 38) for the spacing requirement to be met between Power Plane 2 and Power Plane 1. This template shape is dependent on the voltage between Power Plane 2 and Power Plane 1. The distance between the outer extent of Power Plane 2 and the outer extent of the template shape is explicitly stated in the `electrical_isolation_spacing_requirement` between the Power Plane 2 and Power Plane 1. Populate member of `Electrical_isolation_material_removal_structure` to capture the stated relationships and properties. For further reference in this example, this member will be called 1.



Figure 38 – Power Plane 1 to 2 Material removal template

Step 3: Create a `Material_removal_stratum_feature_template` (see figure 39) for the spacing requirement to be met between Signal 3 and Power Plane 1. This template shape is dependent on the voltage between Signal 3 and Power Plane 1. The distance between the outer extent of Signal 3 and the outer extent of the template shape is explicitly stated in the `electrical_isolation_spacing_requirement` between the Signal 3 and Power Plane 1. Populate a member of `Electrical_isolation_material_removal_structure` to capture the stated relationships and properties. For further reference in the example, this member will be called 2.



Figure 39 – Power Plane 1 to Signal 3 Material removal template

Step 4: Create a `Material_removal_stratum_feature_template` (see figure 40) for the spacing requirement to be met between Signal 3 and Power Plane 2. This template shape is dependent on the voltage between Signal 3 and Power Plane 2. The distance between the outer extent of Signal 3 and the outer extent of the template shape is explicitly stated in the `electrical_isolation_spacing_requirement` between the Signal 3 and Power Plane 2. Populate a member of `Electrical_isolation_material_removal_structure` to capture the stated relationships and properties. For further reference in the example, this member will be called 3. As a summary at this point, all the design intent shapes are defined; all the electrical isolation shapes are defined.



Figure 40 – Power Plane 2 to Signal 3 Material removal template

See figure 41 for the result achieved by a processor in conformance with this part of ISO 10303, in support of a CAD application, or in support of a manufacturing application.

Since Signal 3 has a higher priority than Power Plane 2, the `precedent_modification` attribute specifies `Electrical_isolation_material_removal_structure #1` and the `subsequent_modification` specifies `Electrical_isolation_material_removal_structure #2`

The data associated with a `Design_intent_modification` are the following:

- `design_intent`;
- `modification_causal_item`.

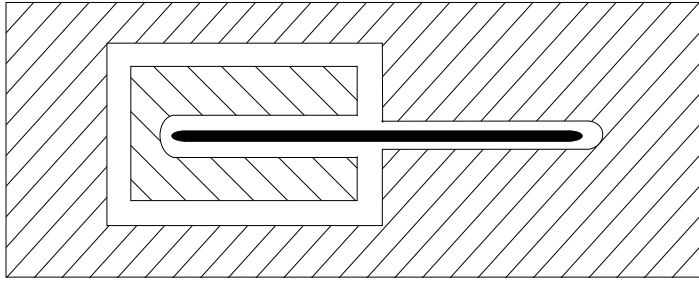


Figure 41 – Final shape for the combined signals

4.2.267.1 design_intent

The `design_intent` specifies the `Stratum_feature` (see 4.2.851) that is changed by the `Design_intent_modification`.

4.2.267.2 modification_causal_item

The `modification_causal_item` specifies one of either an `Assembly_component` (see 4.2.34), a `Component_external_reference` (see 4.2.117), a `Component_feature` (see 4.2.118), or a `Component_feature_external_reference` (see 4.2.119), or one of an `Ee_requirement_occurrence` (see 4.2.330), a `Stratum` (see 4.2.845), or one of an `Inter_stratum_feature` (see 4.2.432), a `Laminate_component` (see 4.2.483), a `Stratum_feature` (see 4.2.851), or a `Stratum_surface` (see 4.2.868) that is the cause of the `Design_intent_modification`.

Constraints:

WR1: A processor reading an exchange file based on this part of ISO 10303 shall interpret the lack of member of `Design_intent_modification_sequence` as meaning that there are no conflicting members of `Electrical_isolation_material_removal_structure` in the exchange file, and that they may be applied in any sequence.

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification
  SUPERTYPE OF (ONEOF (material_removal_feature,
    material_addition_feature, material_shape_modification))
  SUBTYPE OF (managed_design_object);
  design_intent          : stratum_feature;
  modification_causal_item : restriction_basis_item;
END_ENTITY; -- design_intent_modification

```

(*

4.2.268 Design_intent_modification_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_3d_position is a type of Managed_design_object (see 4.2.524). A Design_intent_modification_in_stratum_3d_position is the three dimensional position and orientation of a three dimensional model of a Design_intent_modification (see 4.2.267) in the three dimensional model of the Stratum (see 4.2.845).

The data associated with a Design_intent_modification_3d_position are the following:

- feature_shape;
- stratum_shape;
- transformation.

4.2.268.1 feature_shape

The feature_shape specifies the role of either the Design_intent_modification_3d_shape (see 4.2.269) or of the Design_intent_modification_non_planar_2d_shape (see 4.2.270) for the Design_intent_modification_3d_position.

4.2.268.2 stratum_shape

The stratum_shape specifies the role of the Stratum_3d_shape (see 4.2.846) for the Design_intent_modification_3d_position.

4.2.268.3 transformation

The transformation specifies the role of the Axis_placement_3d (see 4.2.58) for the Design_intent_modification_3d_position.

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_3d_position
  SUBTYPE OF (managed_design_object);
    design_shape : physical_unit_3d_shape;
    transformation : axis_placement_3d;
    feature_shape : design_intent_modification_3d_shape_select;
END_ENTITY; -- design_intent_modification_3d_position

--tt tc1 wd8 changed name, geometric context
( *
```

4.2.269 Design_intent_modification_3d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_3d_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Design_intent_modification_3d_shape is the representation of a Design_intent_modification (see 4.2.267) by a Bound_volume_shape (see 4.2.76).

The data associated with a Design_intent_modification_3d_shape are the following:

- represented_design_intent_modification;
- shape_representation.

4.2.269.1 represented_design_intent_modification

The represented_design_intent_modification specifies the Design_intent_modification (see 4.2.267) that is to be characterized by the Design_intent_modification_3d_shape.

4.2.269.2 shape_representation

The shape_representation specifies the Bound_volume_shape (see 4.2.76) for the representation of the Design_intent_modification (see 4.2.267).

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_3d_shape
  SUBTYPE OF (bound_volume_shape, feature_shape_definition);
    represented_design_intent_modification : design_intent_modification;
(* tt tc1 wd9 deleted
    shape_representation                  : bound_volume_shape;
*)
END_ENTITY; -- design_intent_modification_3d_shape

--tt tc1 wd8 changed name, geometric context
(*
```

4.2.270 Design_intent_modification_non_planar_2d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_non_planar_2d_shape is a type of Managed_design_object (see 4.2.524). A Design_intent_modification_non_planar_2d_shape is the representation of a Design_intent_modification (see 4.2.267) by a Open_shell_based_surface (see 4.2.566).

The data associated with a `Design_intent_modification_non_planar_2d_shape` are the following:

- `represented_design_intent_modification`;
- `shape_representation`.

4.2.270.1 `represented_design_intent_modification`

The `represented_design_intent_modification` specifies the `Design_intent_modification` (see 4.2.267) that is to be characterized by the `Design_intent_modification_non_planar_2d_shape`.

4.2.270.2 `shape_representation`

The `shape_representation` specifies the `Open_shell_based_surface` (see 4.2.566) for the representation of the `Design_intent_modification` (see 4.2.267).

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_non_planar_2d_shape
--tt tc1 wd9 modified
    SUBTYPE OF (open_shell_based_surface);
    represented_design_intent_modification : design_intent_modification;
(* tt tc1 wd9 deleted
    shape_representation                  : open_shell_based_surface;
*)
END_ENTITY; -- design_intent_modification_non_planar_2d_shape

( *
```

4.2.271 `Design_intent_modification_planar_position`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Design_intent_modification_planar_position` is a type of `Managed_design_object` (see 4.2.524). A `Design_intent_modification_in_stratum_planar_position` is the two dimensional position and orientation of a two dimensional model of a `Design_intent_modification` (see 4.2.267) in the two dimensional model of the Stratum (see 4.2.845).

The data associated with a `Design_intent_modification_planar_position` are the following:

- `feature_shape`;
- `stratum_shape`;
- `transformation`.

4.2.271.1 feature_shape

The feature_shape specifies either a Design_intent_modification_planar_shape (see 4.2.272) or a Design_intent_modification_styled_area_representation (see 4.2.274) for the Design_intent_modification_planar_position.

4.2.271.2 stratum_shape

The stratum_shape specifies the role of the Stratum_planar_shape (see 4.2.867) for the Design_intent_modification_planar_position.

4.2.271.3 transformation

The transformation specifies the role of the Axis_placement_2d (see 4.2.57) for the Design_intent_modification_planar_position.

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_planar_position
  SUBTYPE OF (managed_design_object);
    design_shape : physical_unit_planar_shape;
    transformation : axis_placement_2d;
    feature_shape : design_intent_modification_planar_shape_select;
END_ENTITY; -- design_intent_modification_planar_position

( *
```

4.2.272 Design_intent_modification_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_planar_shape is a type of Managed_design_object (see 4.2.524). A Design_intent_modification_planar_shape is the representation of a Design_intent_modification (see 4.2.267) by a Planar_projected_shape (see 4.2.677).

The data associated with a Design_intent_modification_planar_shape are the following:

- represented_design_intent_modification;
- shape_representation.

4.2.272.1 represented_design_intent_modification

The represented_design_intent_modification specifies the Design_intent_modification (see 4.2.267) that is to be characterized by the Design_intent_modification_planar_shape.

4.2.272.2 shape_representation

The shape_surface_representation specifies the Planar_projected_shape (see 4.2.677) for the representation of the Design_intent_modification (see 4.2.267).

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_planar_shape
  SUBTYPE OF (planar_projected_shape);
    represented_design_intent_modification : design_intent_modification;
(* tt tc1 wd9 deleted
    shape_representation                  : planar_projected_shape;
*)
END_ENTITY; -- design_intent_modification_planar_shape

( *
```

4.2.273 Design_intent_modification_sequence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_sequence is a type of Managed_design_object (see 4.2.524). A Design_intent_modification_sequence is a link in a sequence of design modifications that results in the completed design.

The data associated with a Design_intent_modification_sequence are the following:

- precedent_modification;
- subsequent_modification.

4.2.273.1 precedent_modification

The precedent_modification specifies the previous Design_intent_modification (see 4.2.267) or the previous Material_removal_laminate_component (see 4.2.535) for the Design_intent_modification_sequence.

4.2.273.2 subsequent_modification

The subsequent_modification specifies the subsequent Design_intent_modification (see 4.2.267) or the subsequent Material_removal_laminate_component (see 4.2.535) for the Design_intent_modification_sequence.

Constraints:

UR1: The combination of the precedent_modification and the subsequent_modification shall be unique within a population of Design_intent_modification_sequences.

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_sequence
  SUBTYPE OF (managed_design_object);
  precedent_modification :
    design_intent_modification_or_material_removal_component;
  subsequent_modification :
    design_intent_modification_or_material_removal_component;
  UNIQUE
    UR1 : precedent_modification, subsequent_modification;
END_ENTITY; -- design_intent_modification_sequence

( *
```

4.2.274 Design_intent_modification_styled_area_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_intent_modification_styled_area_representation is a type of Managed_design_object (see 4.2.524). A Design_intent_modification_styled_area_representation is the representation of one or more Design_intent_modification (see 4.2.267) as one of a Fill_area_occurrence (see 4.2.377), or a Styled_curve_occurrence (see 4.2.875). A Design_intent_modification_styled_area_representation is parametric data provided to support an algorithm that calculates explicit geometry. Since the geometry may be discontinuous (fill may include dashed lines), more than one Design_intent_modification (see 4.2.267) may be referenced by Design_intent_modification_styled_area_representation as the resulting product data of the algorithm execution. The geometric data resulting from execution of the algorithm shall be included in Design_intent_modification_planar_shape (see 4.2.272) if it is not desired to exchange the Design_intent_modification_styled_area_representation. Exchanging or sharing both the Design_intent_modification_styled_area_representation and the Design_intent_modification_planar_shape that result from execution of the algorithm may be appropriate. Therefore an optional attribute that specifies the planar shapes that are derived from the Design_intent_modification_styled_area_representation is to be included in an exchange if both the planar shape and the styled area representation is included. The units used in the Design_intent_modification_styled_area_representation shall be those specified in the global geometric context for the Interconnect_module (see 4.2.440) design as specified by Cartesian_coordinate_system (see 4.2.86). If the geometry is contiguous, there shall be only one Design_intent_modification referenced.

The data associated with a `Design_intent_modification_styled_area_representation` are the following:

- `derived_planar_shape`;
- `represented_design_intent_modification`;
- `shape_styled_area_representation`.

4.2.274.1 `derived_planar_shape`

The `derived_planar_shape` specifies the `Design_intent_modification_planar_shape` for the `Design_intent_modification_styled_area_representation`. There may be more than one `Design_intent_modification_planar_shape` for the `Design_intent_modification_styled_area_representation`. A `Design_intent_modification_planar_shape` need not be provided for the `Design_intent_modification_styled_area_representation`.

4.2.274.2 `represented_design_intent_modification`

The `represented_design_intent_modification` specifies the `Design_intent_modification` that is characterized by this `Design_intent_modification_styled_area_representation`. There may be more than one `Design_intent_modification` for the `Design_intent_modification_styled_area_representation`.

4.2.274.3 `shape_styled_area_representation`

The `shape_styled_area_representation` specifies one of a `Fill_area_occurrence` (see 4.2.377), or a `Styled_curve_occurrence` (see 4.2.875) that represents the modification in the `Design_intent`.

NOTE EXPRESS specification:

```

*)
ENTITY design_intent_modification_styled_area_representation
  SUBTYPE OF (managed_design_object);
    derived_planar_shape          : OPTIONAL SET [1:?] OF
                                   design_intent_modification_planar_shape;
    represented_design_intent_modification : design_intent_modification;
    shape_styled_area_representation : styled_area_representation;
END_ENTITY; -- design_intent_modification_styled_area_representation

( *
```

4.2.275 `Design_layer_stratum`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Design_layer_stratum` is a type of `Stratum` (see 4.2.845) incorporated into an `Interconnect_module` (see 4.2.440) for the purpose of realizing a physical network in one material.

NOTE `Design_layer_stratum` are usually described as a signal layer or a power or ground plane.

The data associated with a `Design_layer_stratum` are the following:

- `existence_dependence`;
- `of_technology`;
- `primary_design_layer_stratum`.

4.2.275.1 `existence_dependence`

The `existence_dependence` is an inverse relationship that specifies that the existence of the `Design_layer_stratum` is dependent on the existence of the `Layer_connection_point` (see 4.2.499) that specifies the `Design_layer_stratum` as its `resident_design_layer_stratum`. There shall be one or more `Layer_connection_point` for a `Design_layer_stratum`.

4.2.275.2 `of_technology`

The `of_technology` specifies the role of the `Design_layer_technology` (see 4.2.276) for the `Design_layer_stratum`.

4.2.275.3 `primary_design_layer_stratum`

The `primary_design_layer_stratum` specifies whether or not the `Design_layer_stratum` is the `primary_stratum` of the product. The value of `TRUE` indicates that the `Design_layer_stratum` is the `primary_design_layer_stratum`, meaning that it is the stratum occupying the side of the product that has been arbitrarily chosen to be the primary side. The value of `FALSE` indicates that the `Design_layer_stratum` is not the `primary_design_layer_stratum`.

NOTE Usual industry practice is to specify the primary stratum as that which has more components or more active components on it, but this part of ISO 10303 does not enforce that practice.

Constraints:

WR1: The value of `TRUE` for the `primary_design_layer_stratum` shall be specified for exactly one member of `Design_layer_stratum` in the design.

NOTE This constraint is formally defined in the `Interconnect_module` AO.

WR2: The value of `TRUE` for the `primary_design_layer_stratum` shall be specified if and only if the `Design_layer_stratum` is assigned to the `Interconnect_module_primary_surface`. The `Design_layer_stratum` shall be assigned to the `Interconnect_module_primary_surface` if and only if the `primary_design_layer_stratum` is `TRUE`.

NOTE EXPRESS specification:

```

*)
ENTITY design_layer_stratum
  SUBTYPE OF (stratum);
    primary_design_layer_stratum : BOOLEAN;
    SELF\stratum.of_technology    : design_layer_technology;
  INVERSE
    existence_dependence : SET [1:?] OF layer_connection_point FOR
      resident_design_layer_stratum;
WHERE
  WR1: (NOT(primary_design_layer_stratum) AND
    (SIZEOF(QUERY(imps &#60;* USEDIN(SELF,
      'AP210_ARM.STRATUM_CONCEPT_TO_PHYSICAL_USAGE_VIEW_ASSIGNMENT.' +
      'ASSIGNED_DESIGN_OBJECT') |
      ('AP210_ARM.INTERCONNECT_MODULE_PRIMARY_SURFACE' IN
      TYPEOF(imps.defined_part_feature)))) = 0)) OR
    ((primary_design_layer_stratum) AND
    (SIZEOF(QUERY(imps &#60;* USEDIN(SELF,
      'AP210_ARM.STRATUM_CONCEPT_TO_PHYSICAL_USAGE_VIEW_ASSIGNMENT.' +
      'ASSIGNED_DESIGN_OBJECT') |
      ('AP210_ARM.INTERCONNECT_MODULE_PRIMARY_SURFACE' IN
      TYPEOF(imps.defined_part_feature)))) = 1)));
END_ENTITY; -- design_layer_stratum
( *

```

4.2.276 Design_layer_technology

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_layer_technology is a type of Stratum_technology (see 4.2.871). The Design_layer_technology is a technology specification tailored for those materials intended to be used to implement the direct connectivity as described in a design. The same Design_layer_technology may be used in multiple designs for multiple products, since there is no dependence of the Design_layer_technology on the design referencing it.

The data associated with a Design_layer_technology are the following:

- design_layer_purpose;
- minimum_finished_feature_size;
- minimum_finished_feature_spacing.

4.2.276.1 design_layer_purpose

The design_layer_purpose specifies one of the lands_only, other_signal, or power_or_ground purpose of the Design_layer_stratum (see 4.2.275) for the Design_layer_technology.

4.2.276.2 minimum_finished_feature_size

The `minimum_finished_feature_size` specifies the `Length_data_element` (see 4.2.511) of the smallest permissible extent of a feature for the `Design_layer_technology`.

4.2.276.3 minimum_finished_feature_spacing

The `minimum_finished_feature_spacing` specifies the `Length_data_element` (see 4.2.511) of the smallest permissible distance between features for the `Design_layer_technology`.

NOTE EXPRESS specification:

```
*)
ENTITY design_layer_technology
  SUBTYPE OF (stratum_technology);
    design_layer_purpose : layer_purpose;
    SELF\stratum_technology.minimum_finished_feature_size : length_data_element;
    SELF\stratum_technology.minimum_finished_feature_spacing : length_data_element;
END_ENTITY; -- design_layer_technology

(*
```

4.2.277 Padstack_definition

A `Design_layer_type_specific_padstack_definition` is a type of `Padstack_definition` (see 4.2.595) that applies to all design layers in a design to which it is applied.

NOTE EXPRESS specification:

```
*)
ENTITY design_layer_type_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- padstack_definition

(*
```

4.2.278 Design_object_management_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Design_object_management_relationship` is one of an `Add_design_object_management_relationship` (see 4.2.3), a `Change_design_object_management_relationship` (see 4.2.95), or a `Delete_design_object_management_relationship` (see 4.2.257). A `Design_object_management_relationship` is the association between the `Ee_product_definition` (see 4.2.323) changed from to the `Ee_product_definition` changed to.

The data associated with a Design_object_management_relationship are the following:

- current_design;
- previous_design.

4.2.278.1 current_design

The current_design specifies the current Ee_product_definition (see 4.2.323), after the change, for the Design_object_management_relationship.

4.2.278.2 previous_design

The previous_design specifies the Ee_product_definition (see 4.2.323), before the change, for the Design_object_management_relationship.

Constraints:

WR1: The current_design shall not be the previous_design.

WR2: The type of both Ee_product_definitions shall be the same (e.g., both assembly_module).

NOTE EXPRESS specification:

```

*)
ENTITY design_object_management_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (
    change_design_object_management_relationship,
    delete_design_object_management_relationship,
    add_design_object_management_relationship));
  previous_design : ee_product_definition;
  current_design  : ee_product_definition;
END_ENTITY; -- design_object_management_relationship

( *
```

4.2.279 Design_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330). A Design_requirement_occurrence may be an Interface_requirement_occurrence (see 4.2.471).

NOTE EXPRESS specification:

```
*)
ENTITY design_requirement_occurrence
  SUPERTYPE OF (interface_requirement_occurrence)
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- design_requirement_occurrence

( *
```

4.2.280 Design_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_specification is a type of Ee_specification (see 4.2.338) that establishes the design specifications of products. A Design_specification may be an Interface_specification (see 4.2.472). These specifications are not determined by other design or construction features or by referenced data.

NOTE EXPRESS specification:

```
*)
ENTITY design_specification
  SUPERTYPE OF (interface_specification)
  SUBTYPE OF (ee_specification);
END_ENTITY; -- design_specification

( *
```

4.2.281 Design_view_connection_zone_position_relationship

A Design_view_connection_zone_position_relationship is a type of Managed_design_object (see 4.2.524). A Design_view_connection_zone_position_relationship is the association between a Physical_unit_design_view (see 4.2.667), as a target and a Connection_zone (see 4.2.185) as the source. This relationship positions and orients the geometric model of the source into the model of the target using a transformation.

NOTE There can be only one terminal that uses the Connection_zone related so there is no need to reference the terminal in this relationship.

The data associated with a `Design_view_connection_zone_position_relationship` are the following:

- `associating_design_view_shape`;
- `associated_usage`;
- `associated_connection_zone_shape_definition`;
- `associated_usage_placement`;
- `associating_design_view`.

4.2.281.1 associating_design_view_shape

The `associating_design_view_shape` specifies the `Feature_shape_definition` (see 4.2.372), that is the shape that contains the target for the `Design_view_connection_zone_position_relationship`.

4.2.281.2 associated_usage

The `associated_usage` specifies the `Connection_zone` (see 4.2.185), that is the source item for the `Design_view_connection_zone_position_relationship`.

4.2.281.3 associated_connection_zone_shape_definition

The `associated_connection_zone_shape_definition` specifies the `Connection_zone_shape` (see ??), that is the source shape for the `Design_view_connection_zone_position_relationship`.

4.2.281.4 associated_usage_placement

The `associated_usage_placement` specifies the `Axis_placement` (see 4.2.56) that is the target item in the coordinate system of the terminal shape for the `Design_view_connection_zone_position_relationship`, including any transformation required.

4.2.281.5 associating_design_view

The `associating_design_view` specifies the `Physical_unit_design_view` (see 4.2.667) for which the shape is composed for the `Design_view_connection_zone_position_relationship`.

Constraints:

IP1: The `feature_shape_definition` of the `associating_design_view` shall be the `associating_design_view_shape`.

IP2: The `associated_usage connection_zone` shall be a `connection_zone` in the scope of the `associating_design_view`.

NOTE EXPRESS specification:

```

*)
ENTITY design_view_connection_zone_position_relationship
  SUBTYPE OF (managed_design_object);
    associating_design_view_shape          : feature_shape_definition;
    associated_usage                        : connection_zone;
    associated_connection_zone_shape_definition : feature_shape_definition;
    associated_usage_placement              : axis_placement;
    associating_design_view                 : physical_unit_design_view;
END_ENTITY; -- design_view_connection_zone_position_relationship
( *
```

4.2.282 Design_view_terminal_component_shape_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Design_view_terminal_component_shape_relationship is a type of Managed_design_object (see 4.2.524). A Design_view_terminal_component_shape_relationship is the association between a Conductive_interconnect_element_terminal (see 4.2.172), or an Interconnect_component_join_terminal (see 4.2.439), or an Interconnect_component_interface_terminal (see 4.2.438), or a Movable_packaged_component_join_terminal (see 4.2.572) as a source and either a Conductive_interconnect_element (see 4.2.171) or a Physical_component (see 4.2.648) or a Packaged_component (see 4.2.585) as the target, respectively. This relationship positions and orients the geometric model of the source into the model of the target using a transformation. A Conductive_interconnect_element_terminal shall not be associated with a Physical_component or with a Packaged_component. An Interconnect_component_join_terminal shall not be associated with a Conductive_interconnect_element or a Packaged_component. An Interconnect_component_interface_terminal shall not be associated with a Conductive_interconnect_element or a Packaged_component. A Movable_packaged_component_join_terminal shall only be associated with a Packaged_component.

The data associated with a Design_view_terminal_component_shape_relationship are the following:

- associating_component_shape;
- associated_usage;
- associated_terminal_shape_definition;
- associated_usage_placement.

4.2.282.1 associating_component_shape

The associating_component_shape specifies the Feature_shape_definition (see 4.2.372), that is the component shape for the Design_view_terminal_component_shape_relationship.

4.2.282.2 associated_usage

The `associated_usage` specifies the `Interconnect_component_join_terminal` (see 4.2.439), or the `Movable_packaged_component_join_terminal` (see 4.2.572), or the `Conductive_interconnect_element_terminal` (see 4.2.172), or the `Interconnect_component_interface_terminal` (see 4.2.438) that is the source item for the `Design_view_terminal_component_shape_relationship`.

4.2.282.3 associated_terminal_shape_definition

The `associated_terminal_shape_definition` specifies the `Feature_shape_definition` (see 4.2.372) that is the source shape for the `Design_view_terminal_component_shape_relationship`.

4.2.282.4 associated_usage_placement

The `associated_usage_placement` specifies the `Axis_placement` (see 4.2.56) that is the target for the `Design_view_terminal_component_shape_relationship`.

NOTE EXPRESS specification:

```

*)
ENTITY design_view_terminal_component_shape_relationship
  SUBTYPE OF (managed_design_object);
    associating_component_shape          : feature_shape_definition;
    associated_usage                      :
      conductive_interconnect_terminal_or_interconnect_component_join_terminal;
    associated_terminal_shape_definition : feature_shape_definition;
    associated_usage_placement           : axis_placement;
END_ENTITY; -- Design_view_terminal_component_shape_relationship
( *
```

4.2.283 Device_2d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Device_2d_position` is a type of `Managed_design_object` (see 4.2.524). A `Device_2d_position` is the location in two dimensions of a `Part_device` (see 4.2.606) within a `Packaged_part` (see 4.2.591).

The data associated with a `Device_2d_position` are the following:

- `device_shape`;
- `part_shape`;
- `placed_device`;
- `transformation`.

4.2.283.1 device_shape

The device_shape specifies the Physical_unit_planar_shape (see 4.2.670) of the Part_device (see 4.2.606) for the Device_2d_position.

4.2.283.2 part_shape

The part_shape specifies the Physical_unit_planar_shape (see 4.2.670) of the Packaged_part (see 4.2.591) for the Device_2d_position.

4.2.283.3 placed_device

The placed_device specifies the role of the Part_device (see 4.2.606) for the Device_2d_position.

4.2.283.4 transformation

The transformation specifies the role of the Axis_placement_2d (see 4.2.57) for the Device_2d_position.

NOTE EXPRESS specification:

```

*)
ENTITY device_2d_position
  SUBTYPE OF (managed_design_object);
    placed_device : part_device;
    transformation : axis_placement_2d;
    part_shape    : physical_unit_planar_shape;
    device_shape  : physical_unit_planar_shape;
END_ENTITY; -- device_2d_position

```

(*

4.2.284 Device_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Device_3d_position is a type of Managed_design_object (see 4.2.524). A Device_3d_position is the location in three dimensions of a Part_device (see 4.2.606) within a Packaged_part (see 4.2.591).

The data associated with a Device_3d_position are the following:

- device_shape;
- part_shape;
- placed_device;
- transformation.

4.2.284.1 device_shape

The device_shape specifies the Physical_unit_3d_shape (see 4.2.666) of the Part_device (see 4.2.606) for the Device_3d_position.

4.2.284.2 part_shape

The part_shape specifies the Physical_unit_3d_shape (see 4.2.666) of the Packaged_part (see 4.2.591) for the Device_3d_position.

4.2.284.3 placed_device

The placed_device specifies the role of the Part_device (see 4.2.606) for the Device_3d_position.

4.2.284.4 transformation

The transformation specifies the role of the Axis_placement_3d (see 4.2.58) for the Device_3d_position.

NOTE EXPRESS specification:

```

*)
ENTITY device_3d_position
  SUBTYPE OF (managed_design_object);
    placed_device : part_device;
    transformation : axis_placement_3d;
    part_shape : physical_unit_3d_shape;
    device_shape : physical_unit_3d_shape;
END_ENTITY; -- device_3d_position

```

(*

4.2.285 Device_terminal_map

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Device_terminal_map is a type of Managed_design_object (see 4.2.524). A Device_terminal_map is the association between a Part_device_terminal (see 4.2.607) and a Packaged_part_terminal (see 4.2.594). A Device_terminal_map provides the topological equivalent of a bond wire from the package terminal to the terminal of the installed Part_device (see 4.2.606).

The data associated with a Device_terminal_map are the following:

- mapped_device_terminal;
- target_terminal.

4.2.285.1 mapped_device_terminal

The mapped_device_terminal specifies the Part_device_terminal (see 4.2.607) from which the association is made to the target_terminal by the Device_terminal_map.

4.2.285.2 target_terminal

The target_terminal specifies the targeted Packaged_part_terminal (see 4.2.594) for the Device_terminal_map.

Constraints:

UR1: The combination of the target_terminal and mapped_device_terminal shall be unique within a population of Device_terminal_map.

WR1: The mapped_device_terminal and the target_terminal shall be in the context of the same Packaged_part.

NOTE Reference Part_device_terminal_model_port_assignment for Analytical_model_port_assignment.

NOTE EXPRESS specification:

```
*)
ENTITY device_terminal_map
  SUBTYPE OF (managed_design_object);
    target_terminal      : packaged_part_terminal;
    mapped_device_terminal : part_device_terminal;
  UNIQUE
    UR1 : mapped_device_terminal, target_terminal;
END_ENTITY; -- device_terminal_map

(*
```

4.2.286 Device_terminal_map_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Device_terminal_map_relationship is a type of Managed_design_object (see 4.2.524). A Device_terminal_map_relationship is the association of one Device_terminal_map (see 4.2.285) to another.

The data associated with a Device_terminal_map_relationship are the following:

- related_device_terminal_map;
- relating_device_terminal_map.

4.2.286.1 related_device_terminal_map

The related_device_terminal_map specifies the target Device_terminal_map (see 4.2.285) in the Device_terminal_map_relationship.

4.2.286.2 relating_device_terminal_map

The relating_device_terminal_map specifies the source Device_terminal_map (see 4.2.285) in the Device_terminal_map_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY device_terminal_map_relationship
  SUBTYPE OF (managed_design_object);
    related_device_terminal_map : device_terminal_map;
    relating_device_terminal_map : device_terminal_map;
END_ENTITY; -- device_terminal_map_relationship

( *
```

4.2.287 Diametrical_boundary_size_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Diametrical_boundary_size_characteristic is a type of Boundary_size_characteristic (see 4.2.78) that defines the diameter of a Tolerance_zone_circular_or_cylindrical_or_spherical_boundary (see 4.2.919).

The data associated with a Diametrical_boundary_size_characteristic are the following:

— sized_boundary.

4.2.287.1 sized_boundary

The sized_boundary specifies the Tolerance_zone_circular_or_cylindrical_or_spherical_boundary (see 4.2.919), the diameter of which is defined by the Diametrical_boundary_size_characteristic.

Constraints:

WR1: The Tolerance_zone_boundary specified by the sized_boundary shall be the same instance of Tolerance_zone_boundary as specified by the defined_zone_boundary of the Boundary_zone_definition_with_specified_size that is specified as the boundary_definition.

NOTE The attribute boundary_definition is declared in the Boundary_size_characteristic entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY diametrical_boundary_size_characteristic
  SUBTYPE OF (boundary_size_characteristic);
    sized_boundary : tolerance_zone_circular_or_cylindrical_or_spherical_boundary;
END_ENTITY; -- diametrical_boundary_size_characteristic

( *

```

4.2.288 Dielectric_crossover_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dielectric_crossover_area is a type of Stratum_concept_relationship (see 4.2.849) that relates one Stratum_feature (see 4.2.851) with another that results in the creation of a dielectric crossover feature in a dielectric Stratum (see 4.2.845).

The data associated with a Dielectric_crossover_area are the following:

- related_concept;
- relating_concept;
- resulting_dielectric_crossover.

4.2.288.1 related_concept

The related_concept specifies the source Stratum_feature (see 4.2.851) for the Dielectric_crossover_area.

4.2.288.2 relating_concept

The relating_concept specifies the target Stratum_feature (see 4.2.851) for the Dielectric_crossover_area.

4.2.288.3 resulting_dielectric_crossover

The resulting_dielectric_crossover specifies the Stratum_feature (see 4.2.851) in a stratum of dielectric material for the Dielectric_crossover_area.

Constraints:

WR1: The Stratum on which the Stratum_features reside related to this Dielectric_crossover_area shall be arranged such that the resulting_dielectric_crossover is between the relating_concept and the related_concept.

NOTE There is no directionality implied by the use of "related", "relating" in the attribute names.

WR2: The relating_concept shall not be the related_concept.

NOTE EXPRESS specification:

```
*)
ENTITY dielectric_crossover_area
  SUBTYPE OF (stratum_concept_relationship);
    related_concept          : stratum_feature;
    relating_concept         : stratum_feature;
    resulting_dielectric_crossover : stratum_feature;
END_ENTITY; -- dielectric_crossover_area

( *
```

4.2.289 Dielectric_material_passage

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dielectric_material_passage is a type of Inter_stratum_feature (see 4.2.432) that contains dielectric material. The extent of the Dielectric_material_passage is between the boundary defined by the shape_definition of this Dielectric_material_passage and the shape of the Cutout that is to be filled with dielectric material during manufacture.

EXAMPLE 1 Figure 42 illustrates an exploded, stylized, detail of a Dielectric_material_passage as it appears in two sequential stages of the fabrication process.

The data associated with a Dielectric_material_passage are the following:

- feature_material;
- precedent_passage.

EXAMPLE 2 A Dielectric_material_passage is used, for example, in printed circuit boards with metal core Stratum for thermal conductivity, that have oversized passages created through the metal Stratum before bonding. These oversized passages in the metal Stratum are filled with dielectric material during the bonding process for later machining to suit the eventual purpose for the passage at that location. These smaller passages are often plated for electrical continuity among Design_layer_stratums on either side of the metal core, but there must be electrical isolation between the plating in the passage and the metal thermal conductive Stratum. The dielectric fill material provides that isolation.

4.2.289.1 feature_material

The feature_material specifies either the Ee_material (see 4.2.318) or the Ee_requirement_occurrence (see 4.2.330) that defines the dielectric material contained in the Dielectric_material_passage. The feature_material need not be specified for a particular Dielectric_material_passage.

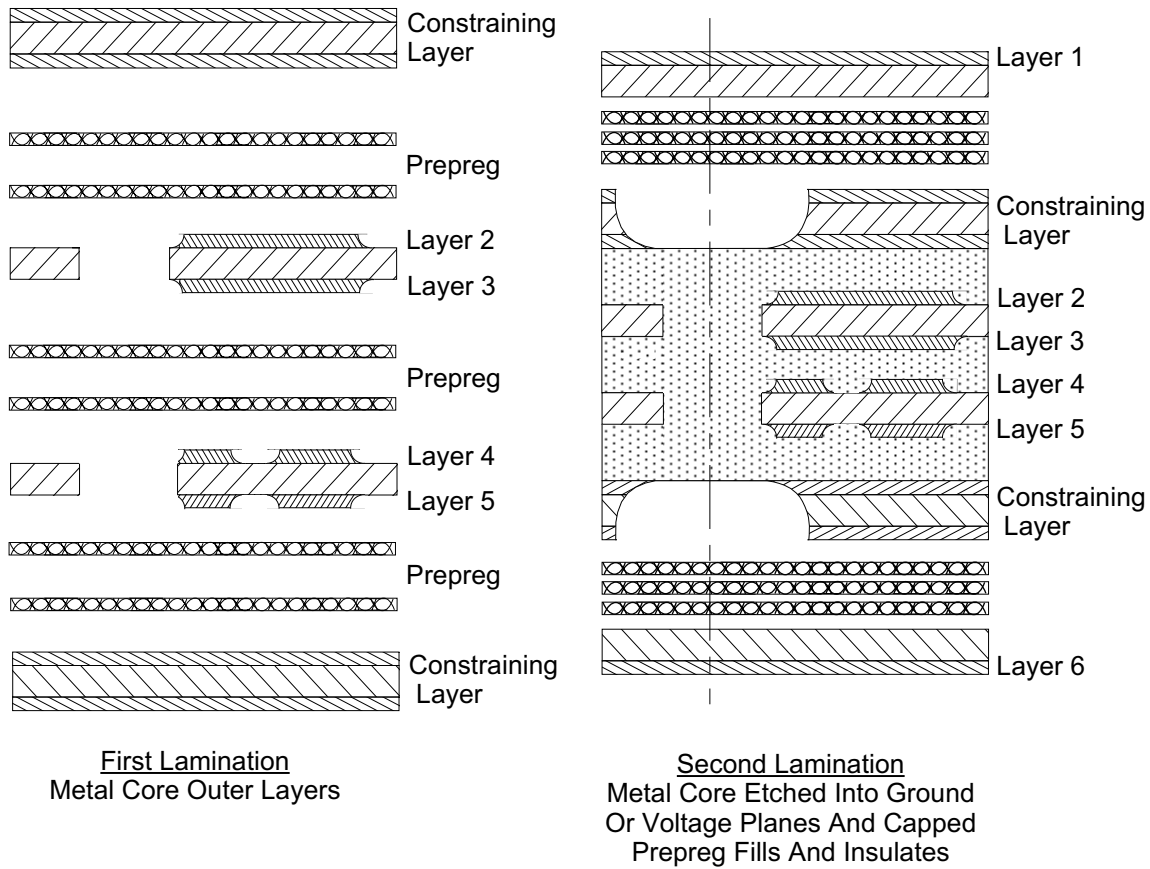


Figure 42 – Dielectric_material_passage exploded view

4.2.289.2 precedent_passage

The precedent_passage specifies the Cutout (see 4.2.208) that defines the outer boundary of the Dielectric_material_passage before it is filled with the specified non-conductive Ee_material (see 4.2.318).

Constraints:

WR1: The same material shall exist throughout the extent of the passage.

NOTE EXPRESS specification:

```
*)
ENTITY dielectric_material_passage
  SUBTYPE OF (inter_stratum_feature);
    feature_material : OPTIONAL material_or_specification;
    precedent_passage : cutout;
END_ENTITY; -- dielectric_material_passage
```

(*

4.2.290 Digital_analytical_model_port

A Digital_analytical_model_port is a type of Analytical_model_port (see 4.2.17) that defines a digital context.

The data associated with a Digital_analytical_model_port are the following:

— nominal_signal_flow_direction.

4.2.290.1 nominal_signal_flow_direction

The nominal_signal_flow_direction specifies either BIDIRECTIONAL, INPUT_DIRECTION, NOT_APPLICABLE, OUTPUT_DIRECTION, or UNKNOWN_DIRECTION for the Digital_analytical_model_port.

Constraints:

WR1: "NOT_APPLICABLE" is not an allowed value.

NOTE EXPRESS specification:

```
*)
ENTITY digital_analytical_model_port
  SUBTYPE OF (analytical_model_port);
WHERE
  WR1: not_applicable &#60;&#62;
    SELF\analytical_model_port.nominal_signal_flow_direction;
END_ENTITY; -- digital_analytical_model_port
```

(*

4.2.291 Dimension_along_curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimension_along_curve` is a type of `Dimensional_location` (see 4.2.300) that defines the location of a `Shape_element` (see 4.2.810) in relationship to another `Shape_element` and in which a `Shape_element` is specified as the path along which the `Dimension_along_curve` characterizes this relationship.

The data associated with a `Dimension_along_curve` are the following:

— `path`.

4.2.291.1 `path`

The path specifies the `Shape_element` (see 4.2.810) along which the relationship between the two `Shape_element`s is characterized by the `Dimension_along_curve`.

Constraints:

UR1: The combination of `located_elements` inherited from `Dimensional_location` and `path` shall be unique within a population of `Dimension_along_curve`.

WR1: The `dimension_type` of the `Dimension_location_with_direction_vector` shall be either "arc_length", or "linear".

NOTE The attribute `dimension_type` is declared in the `Dimensional_characteristic` entity of which this entity is a subtype.

NOTE EXPRESS specification:

```
*)
ENTITY dimension_along_curve
  SUBTYPE OF (dimensional_location);
  path : shape_element;
  UNIQUE
    UR1 : located_elements, path;
END_ENTITY; -- dimension_along_curve
```

(*

4.2.292 `Dimension_location_with_direction_vector`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimension_location_with_direction_vector` is a type of `Dimensional_location` (see 4.2.300) in which the characterization of the relationship between the two `Shape_element`s (see 4.2.810) is dependent on a vector that establishes a measurement direction.

The data associated with a `Dimension_location_with_direction_vector` are the following:

— `measurement_direction`.

4.2.292.1 `measurement_direction`

The `measurement_direction` either specifies the `Datum_based_vector_orientation` (see 4.2.220) or the `Axis_placement` (see 4.2.56) that establishes the measurement direction. If the `measurement_direction` specifies a `Datum_based_vector_orientation`, the measurement direction is established from Datum (see 4.2.214). If the `measurement_direction` specifies an `Axis_placement`, the measurement direction is based on the `Axis_placement` and the representation of the `Shape_elements` whose relationship is characterized by the `Dimension_location_with_direction_vector`.

Constraints:

UR1: The combination of `located_elements` inherited from `Dimensional_location` and `measurement_direction` shall be unique within a population of `Dimension_location_with_direction_vector`.

NOTE The attribute `located_elements` is declared in the `Dimensional_location` entity of which this entity is a subtype.

WR1: The `dimension_type` of the `Dimension_location_with_direction_vector` shall be "linear".

NOTE The attribute `dimension_type` is declared in the `Dimensional_characteristic` entity of which this entity is a subtype.

IP1: If the `measurement_direction` specifies an `Axis_placement`, the `Axis_placement` shall be in the same context as the representation of the two `Shape_elements` that are located relative to each other by the `Dimension_location_with_direction_vector`.

NOTE EXPRESS specification:

```
*)
ENTITY dimension_location_with_direction_vector

    SUBTYPE OF (dimensional_location);
        measurement_direction : axis_placement_or_datum_based_vector_orientation_select;
    UNIQUE
        UR1 : located_elements, measurement_direction;
END_ENTITY; -- dimension_location_with_direction_vector

( *
```

4.2.293 `Dimension_location_with_specified_datum_system`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimension_location_with_specified_datum_system` is a type of `Dimensional_location` (see 4.2.300) that is related to a `Datum_reference_frame` (see 4.2.235).

The data associated with a `Dimension_location_with_specified_datum_system` are the following:

— `referenced_datum_system_defining_conditions`.

4.2.293.1 `referenced_datum_system_defining_conditions`

The `referenced_datum_system_defining_conditions` specifies the `Datum_system_defining_conditions_without_material_conditions` (see 4.2.239) that defines the order of precedence for the datums of a `Datum_system` (see 4.2.236) which in turn establishes the `Datum_reference_frame` (see 4.2.235) to which the `Dimension_location_with_specified_datum_system` is related.

NOTE The specification of the relationship between linear and angular dimensions to a datum reference frame is discussed in 4.4 of ASME Y14.5M.

NOTE EXPRESS specification:

```
*)
ENTITY dimension_location_with_specified_datum_system
  SUBTYPE OF (dimensional_location);
    referenced_datum_system_defining_conditions :
    datum_system_defining_conditions_without_material_conditions;
END_ENTITY; -- dimension_location_with_specified_datum_system

( *
```

4.2.294 `Dimension_qualifier`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimension_qualifier` is a type of `Managed_design_object` (see 4.2.524). A `Dimension_qualifier` is either a `Pre_defined_dimension_qualifier` (see 4.2.707) or a `User_defined_dimension_qualifier` (see 4.2.944). A `Dimension_qualifier` is a description that qualifies one or more `Dimensional_characteristic` (see 4.2.298).

NOTE Qualifying notes are described in 5.3 of ISO/DIS 1101.

The data associated with a `Dimension_qualifier` are the following:

— `qualified_dimensions`.

4.2.294.1 qualified_dimensions

The qualified_dimensions specifies the Dimensional_characteristic (see 4.2.298) to which the Dimension_qualifier applies. There shall be one or more Dimensional_characteristic specified as the qualified_dimensions for a Dimension_qualifier.

NOTE EXPRESS specification:

```
*)
ENTITY dimension_qualifier
  ABSTRACT SUPERTYPE OF (ONEOF (pre_defined_dimension_qualifier,
    user_defined_dimension_qualifier))
  SUBTYPE OF (managed_design_object);
  qualified_dimensions : SET [1:?] OF dimensional_characteristic;
END_ENTITY; -- dimension_qualifier

( *
```

4.2.295 Dimension_related_positional_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimension_related_positional_boundary is a type of Positional_boundary (see 4.2.698) that is a theoretical boundary that is composed of a set of one or more Positional_boundary_member (see 4.2.701). Each Positional_boundary_member is defined by a separate Boundary_based_position_tolerance (see 4.2.77) in conjunction with a Dimensional_size (see 4.2.301).

The data associated with a Dimension_related_positional_boundary are the following:

- boundary_members;
- constrained_physical_feature.

4.2.295.1 boundary_members

The boundary_members specifies the Positional_boundary_member (see 4.2.701) that comprise the Dimension_related_positional_boundary. There shall be one or more Positional_boundary_member specified as the boundary_members for the Dimension_related_positional_boundary.

4.2.295.2 constrained_physical_feature

The constrained_physical_feature specifies the Physical_feature_or_part_template (see 4.2.658) that is to be constrained by the Dimension_related_positional_boundary.

Constraints:

WR1: The Positional_boundary_definition specified as the boundary_definition shall be a Dimension_related_positional_boundary_definition.

NOTE The attribute boundary_definition is declared in the Positional_boundary entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY dimension_related_positional_boundary
  SUBTYPE OF (positional_boundary);
    boundary_members      : SET [1:?] OF positional_boundary_member;
    constrained_physical_feature : physical_feature_or_part_template;
END_ENTITY; -- dimension_related_positional_boundary

( *

```

4.2.296 Dimension_related_positional_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimension_related_positional_boundary_definition is a type of Positional_boundary_definition (see 4.2.700) that characterizes a Dimension_related_positional_boundary (see 4.2.295).

The data associated with a Dimension_related_positional_boundary_definition are the following:

- characterized_boundary_member_definitions;
- defined_positional_boundary.

4.2.296.1 characterized_boundary_member_definitions

The characterized_boundary_member_definitions specifies the Positional_boundary_member_definition (see 4.2.702) that characterize each Positional_boundary_member (see 4.2.701) that comprise the Dimension_related_positional_boundary (see 4.2.295). There shall be one or more Positional_boundary_member_definition specified as the characterized_boundary_member_definitions for the Dimension_related_positional_boundary_definition.

4.2.296.2 defined_positional_boundary

The defined_positional_boundary specifies the Dimension_related_positional_boundary (see 4.2.295) that is characterized by the Dimension_related_positional_boundary_definition.

Constraints:

WR1: Each `Boundary_based_position_tolerance` that defines a `Positional_boundary_member_definition` of the `Positional_boundary_definition` shall reference the same instance of `Datum_system_defining_conditions`.

WR2: Each `Boundary_based_position_tolerance` that defines a `Positional_boundary_member_definition` of the `Positional_boundary_definition` shall have the same value for the `applied_material_condition_property_name`.

NOTE EXPRESS specification:

```

*)
ENTITY dimension_related_positional_boundary_definition
  SUBTYPE OF (positional_boundary_definition);
  SELF\positional_boundary_definition.defined_positional_boundary :
dimension_related_positional_boundary;
  characterized_boundary_member_definitions :
SET [1:?] OF positional_boundary_member_definition;
END_ENTITY; -- dimension_related_positional_boundary_definition

( *
```

4.2.297 Dimension_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimension_set` is a type of `Managed_design_object` (see 4.2.524). A `Dimension_set` is a set of `Dimensional_characteristic` (see 4.2.298) that specify the theoretically exact shape of a `Physical_feature_or_part_template` (see 4.2.658).

The data associated with a `Dimension_set` are the following:

— `associated_dimensions`.

4.2.297.1 associated_dimensions

The `associated_dimensions` specifies the `Dimensional_characteristic` (see 4.2.298) that are associated with the `Dimension_set`. There shall be one or more `Dimensional_characteristic` specified as the `associated_dimensions` for the `Dimension_set`.

Constraints:

WR1: The `Dimension_set` shall be specified as the `profile_representation` by at least one `True_profile`.

NOTE EXPRESS specification:

```

*)
ENTITY dimension_set
  SUBTYPE OF (managed_design_object);
  associated_dimensions : SET [1:?] OF dimensional_characteristic;
END_ENTITY; -- dimension_set

( *

```

4.2.298 Dimensional_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimensional_characteristic is a type of Managed_design_object (see 4.2.524). A Dimensional_characteristic is either a Dimensional_location (see 4.2.300) or a Dimensional_size (see 4.2.301) that characterizes either the spatial constraints between two Shape_element (see 4.2.810) or the size of a Shape_element. Additionally, a Dimensional_characteristic may be a Statistical_dimensional_tolerance (see 4.2.840).

The data associated with a Dimensional_characteristic are the following:

— dimension_type.

4.2.298.1 dimension_type

The dimension_type specifies the label by which the Dimensional_characteristic is known. The dimension_type shall be one of "angular", "arc_length", "controlled_radius", "diameter", "linear", or "radius".

NOTE A description of controlled radius is given in 2.15.2 of ASME Y14.5M.

Constraints:

WR1: The Dimensional_characteristic shall be specified as the dimension by at most one Dimensional_characteristic_representation that specifies a value of FALSE for the auxiliary_dimension_representation.

WR2: If the dimension_type is "angular" and the Dimensional_characteristic is represented by a Shape_dimension_representation that specifies one or more Data_element, each Data_element shall be an Angle_data_element.

WR3: The Dimensional_characteristic shall either be a Dimensional_size or a Dimensional_location.

WR4: If the dimension_type is other than "angular" and the Dimensional_characteristic is represented by a Shape_dimension_representation that specifies one or more Data_element, each Data_element shall be a Length_data_element.

WR5: There shall be exactly one Ee_requirement_occurrence with the description of "Geometric Dimensioning and Tolerancing Reference" that specifies at least one Ee_specification as the requirement_specification.

IP1: There shall be exactly one Ee_requirement_occurrence with the description of "Geometric Dimensioning and Tolerancing Reference" that specifies at least one Ee_specification as the requirement_specification. The Ee_specification shall reference those standards by which the geometric dimensioning and tolerancing data is to be interpreted.

NOTE EXPRESS specification:

```

*)
ENTITY dimensional_characteristic
  ABSTRACT SUPERTYPE OF (ONEOF (dimensional_location,dimensional_size)
    ANDOR statistical_dimensional_tolerance)
  SUBTYPE OF (managed_design_object);
    dimension_type : dimensional_types;
END_ENTITY; -- dimensional_characteristic

( *
```

4.2.299 Dimensional_characteristic_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimensional_characteristic_representation is a type of Managed_design_object (see 4.2.524). A Dimensional_characteristic_representation is an association between a Dimensional_characteristic (see 4.2.298) and the Shape_dimension_representation (see 4.2.809) that represents the Dimensional_characteristic.

The data associated with a Dimensional_characteristic_representation are the following:

- auxiliary_dimension_representation;
- dimension;
- representation.

4.2.299.1 auxiliary_dimension_representation

The auxiliary_dimension_representation specifies the boolean value that indicates whether the associated Shape_dimension_representation (see 4.2.809) is to be considered a requirement for the associated Dimensional_characteristic (see 4.2.298) or is specified for informational purposes only. A value of TRUE indicates that the associated Shape_dimension_representation is for informational purposes only. A value of FALSE indicates that the associated Shape_dimension_representation is to be considered a requirement.

NOTE Auxiliary dimension is defined in 3.1.1.3 of ISO 129.

4.2.299.2 dimension

The dimension specifies either a Dimensional_location (see 4.2.300) or a Dimensional_size (see 4.2.301) that is represented by the specified Shape_dimension_representation (see 4.2.809) for the Dimensional_characteristic_representation.

4.2.299.3 representation

The representation specifies the Shape_dimension_representation (see 4.2.809) that represents the specified Dimensional_characteristic (see 4.2.298) for the Dimensional_characteristic_representation.

NOTE EXPRESS specification:

```

*)
ENTITY dimensional_characteristic_representation
  SUBTYPE OF (managed_design_object);
    representation          : shape_dimension_representation;
    dimension                : dimensional_location_or_dimensional_size;
    auxiliary_dimension_representation : BOOLEAN;
END_ENTITY; -- dimensional_characteristic_representation

( *
```

4.2.300 Dimensional_location

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimensional_location is a type of Dimensional_characteristic (see 4.2.298) that defines the location of a Shape_element (see 4.2.810) relative to another Shape_element. A Dimensional_location may be one of an Angular_dimension_with_direction_vector (see 4.2.24), a Dimension_along_curve (see 4.2.291), or a Dimension_location_with_direction_vector (see 4.2.292). Additionally, the Dimensional_location may be a Dimension_location_with_specified_datum_system (see 4.2.293).

The data associated with a Dimensional_location are the following:

- directed;
- located_elements.

4.2.300.1 directed

The directed specifies the boolean value that indicates whether there is significance to the direction of measure of the Dimensional_location. A value of TRUE indicates that the measurement direction shall be from the Shape_element (see 4.2.810) specified as the locating_shape_element to the Shape_element specified as the located_shape_element by the Shape_element_locating_relationship (see 4.2.815) that is characterized by the Dimensional_location. A value of FALSE indicates that there is no significance placed on the direction of measure.

4.2.300.2 located_elements

The located_elements specifies the Shape_element_locating_relationship (see 4.2.815) that specifies the two Shape_elements for which the relationship is characterized by the Dimensional_location.

Constraints:

WR1: The dimension_type of the Dimensional_location shall not be "diameter".

NOTE The attribute dimension_type is declared in the Dimensional_characteristic entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY dimensional_location
  SUPERTYPE OF (dimension_location_with_specified_datum_system ANDOR
    ONEOF (dimension_along_curve,
      angular_dimension_with_direction_vector,
      dimension_location_with_direction_vector))
  SUBTYPE OF (dimensional_characteristic);
    located_elements : shape_element_locating_relationship;
    directed          : BOOLEAN;
END_ENTITY; -- dimensional_location

( *
```

4.2.301 Dimensional_size

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dimensional_size is a type of Dimensional_characteristic (see 4.2.298) that defines the size of a Shape_element (see 4.2.810). A Dimensional_size is either a Dimensional_size_based_on_opposing_boundaries (see 4.2.302), or a Single_boundary_dimension (see 4.2.827). Additionally, a Dimensional_size may be an Angular_size_dimension (see 4.2.25).

The data associated with a `Dimensional_size` are the following:

- `envelope`;
- `size_of`.

4.2.301.1 `envelope`

The `envelope` specifies the boolean value that indicates whether or not the `Shape_element` (see 4.2.810) for which the size is characterized by the `Dimensional_size` has the additional requirement that it shall not violate the envelope of perfect form at the maximum material size of the `Shape_element`. A value of `TRUE` indicates that the requirement is in effect. A value of `FALSE` indicates that the requirement is not in effect.

NOTE The envelope requirement is described in 6.1 of ISO 8015.

4.2.301.2 `size_of`

The `size_of` specifies the `Shape_element` (see 4.2.810) for which the size is characterized by the `Dimensional_size`.

NOTE EXPRESS specification:

```

*)
ENTITY dimensional_size
  ABSTRACT SUPERTYPE OF (ONEOF (
    dimensional_size_based_on_opposing_boundaries,
    single_boundary_dimension) ANDOR angular_size_dimension)
  SUBTYPE OF (dimensional_characteristic);
  size_of : shape_element;
  envelope : BOOLEAN;
END_ENTITY; -- dimensional_size

( *
```

4.2.302 `Dimensional_size_based_on_opposing_boundaries`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Dimensional_size_based_on_opposing_boundaries` is a type of `Dimensional_size` (see 4.2.301) that characterizes the size of a `Composite_unit_shape_element` (see 4.2.167) as well as characterizing the relationship between two opposing `Shape_element` (see 4.2.810) from which the `Composite_shape_element` (see 4.2.162) is composed.

The data associated with a Dimensional_size_based_on_opposing_boundaries are the following:

- boundary_separation;
- size_of.

4.2.302.1 boundary_separation

The boundary_separation specifies the Shape_element_locating_relationship (see 4.2.815) that specifies the two Shape_elements that are on the boundary of the Composite_unit_shape_element (see 4.2.167) for which the separation is characterized by the Dimensional_size_based_on_opposing_boundaries and is the basis for the size of the Composite_unit_shape_element.

4.2.302.2 size_of

The size_of specifies the Composite_unit_shape_element (see 4.2.167) for which the size is characterized by the Dimensional_size_based_on_opposing_boundaries.

NOTE The attribute size_of is declared in the Dimensional_size (see 4.2.301) entity of which this entity is a subtype.

EXAMPLE 1 The size of a Shape_element (see 4.2.810) that consists of two opposing surfaces of a notch would be characterized by a Dimensional_size_based_on_opposing_boundaries.

Constraints:

WR1: The Shape_element that is specified as the located_shape_element by the Shape_element_locating_relationship specified as the boundary_separation shall be specified as the related_shape_element by a Shape_element_composing_relationship that specifies the Composite_unit_shape_element that is specified as the size_of as the relating_shape_element.

WR2: The Shape_element that is specified as the locating_shape_element by the Shape_element_locating_relationship specified as the boundary_separation shall be specified as the related_shape_element by a Shape_element_composing_relationship that specifies the Composite_unit_shape_element that is specified as the size_of as the relating_shape_element.

WR3: The dimension_type shall be either "angular" or "linear".

NOTE The attribute dimension_type is declared in the Dimensional_characteristic entity of which this entity is a subtype.

IP1: The two shape elements that correspond to the Shape_elements that compose the Composite_unit_shape_element to which the Dimensional_size_based_on_opposing_boundaries is applied shall be opposite each other.

NOTE EXPRESS specification:

```

*)
ENTITY dimensional_size_based_on_opposing_boundaries
  SUBTYPE OF (dimensional_size);
    boundary_separation      : shape_element_locating_relationship;
    SELF\dimensional_size.size_of : composite_unit_shape_element;
END_ENTITY; -- dimensional_size_based_on_opposing_boundaries

( *
```

4.2.303 Directed_axis

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Directed_axis is a type of Non_feature_shape_element (see 4.2.560) that is an axis in which a direction is defined. A Directed_axis is defined by a base Centre_axis (see 4.2.90), with which the Directed_axis is congruent, and two Shape_element (see 4.2.810) through which the base Centre_axis passes or would pass if the Shape_elements were extended. One of the two Shape_elements is specified as the start_shape_element and the other one specified as the end_shape_element.

The data associated with a Directed_axis are the following:

- axis_to_be_oriented;
- end_shape_element;
- start_shape_element.

4.2.303.1 axis_to_be_oriented

The axis_to_be_oriented specifies the Centre_axis (see 4.2.90) that serves as a base axis by which the Directed_axis is defined.

4.2.303.2 end_shape_element

The end_shape_element specifies the Shape_element (see 4.2.810) through which the Directed_axis passes or would pass if the Shape_element were extended. The Shape_element specified by the end_shape_element is in the positive direction from the Shape_element specified by the start_shape_element.

4.2.303.3 start_shape_element

The start_shape_element specifies the Shape_element (see 4.2.810) through which the Directed_axis passes or would pass if the Shape_element were extended. The Shape_element specified by the start_shape_element is in the negative direction from the Shape_element specified by the end_shape_element.

Constraints:

UR1: The combination of `axis_to_be_oriented` and `start_shape_element` shall be unique within a population of `Directed_axis`.

WR1: The `Shape_element` specified by the `start_shape_element` shall not be the same `Shape_element` as specified by the `end_shape_element`.

IP1: The base `Centre_axis` shall pass through or would pass through the `Shape_elements` specified by the `start_shape_element` and `end_shape_element` if the `Shape_elements` were extended.

NOTE EXPRESS specification:

```
*)
ENTITY directed_axis
  SUBTYPE OF (non_feature_shape_element);
    axis_to_be_oriented : centre_axis;
    start_shape_element : shape_element;
    end_shape_element   : shape_element;
  UNIQUE
    UR1 : axis_to_be_oriented, start_shape_element;
END_ENTITY; -- directed_axis
```

(*

4.2.304 Direction

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Direction` is a type of `Managed_design_object`. A `Direction` shall be interpreted as a general direction vector in two-dimensional or three-dimensional space. The actual magnitudes of the components of the vector have no significance; only the ratios among the components are significant.

NOTE This `Application_object` does not provide the details of direction.

NOTE EXPRESS specification:

```
*)
ENTITY direction
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- direction
( *
```

4.2.305 Direction_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Direction_element` is a type of `Managed_design_object` (see 4.2.524). A `Direction_element` is a definition of a shape element that has the form of a half-plane and is used in conjunction with the definition of another half-plane shape element to define a `Datum_based_vector_orientation` (see 4.2.220). In this instance a half-plane has the form of a plane except it is bounded on one side by a straight line. The half-plane shape element that the `Direction_element` defines is oriented with respect to three `Datum_plane` (see 4.2.232).

The data associated with a `Direction_element` are the following:

- `angle_value`;
- `base_datum_plane`;
- `direction`;
- `direction_of_half_datum_plane`;
- `elements_of`.

4.2.305.1 `angle_value`

The `angle_value` specifies the `Angle_data_element` (see 4.2.23) that represents the magnitude of the angle between the half-plane shape element defined by the `Direction_element` and half of the `Datum_plane` (see 4.2.232) specified as the `base_datum_plane`.

4.2.305.2 `base_datum_plane`

The `base_datum_plane` specifies the `Datum_plane` (see 4.2.232) from which the half-plane shape element defined by the `Direction_element` is oriented.

4.2.305.3 `direction`

The `direction` specifies the `Datum_based_direction_vector` (see 4.2.218) that defines the orientation of the angle between the half-plane shape element defined by the `Direction_element` and half of the `Datum_plane` (see 4.2.232) specified by the `base_datum_plane`. The angle is measured from half of the base `Datum_plane` to the defined half-plane shape element in a clockwise direction when viewed in the direction of the `Datum_based_direction_vector`.

4.2.305.4 `direction_of_half_datum_plane`

The `direction_of_half_datum_plane` specifies the `Datum_based_direction_vector` (see 4.2.218) of a `Datum_plane` (see 4.2.232) that intersects the base `Datum_plane` and indicates the half of the base `Datum_plane` from which the half-plane shape element defined by the `Direction_element` is oriented.

4.2.305.5 elements_of

The elements_of is an inverse relationship that specifies that the existence of the Direction_element is dependent on the existence of one or more Datum_based_vector_orientation (see 4.2.220) that specifies the Direction_element as one of their direction_components.

Constraints:

WR1: The Datum_plane specified as the base_datum_plane, the Datum_plane specified as the vector_of by the Datum_based_direction_vector that is specified as the direction, the Datum_plane specified as the vector_of by the Datum_based_direction_vector that is specified as the direction_of_half_datum_plane shall be different.

WR2: The Datum_plane specified as the base_datum_plane, and the Datum_plane specified as the vector_of by the Datum_based_direction_vector that is specified as the direction, and the Datum_plane specified as the vector_of by the Datum_based_direction_vector that is specified as the direction_of_half_datum_plane shall be Datum_planes of the same Datum_reference_frame.

NOTE EXPRESS specification:

```

*)
ENTITY direction_element
  SUBTYPE OF (managed_design_object);
    direction          : datum_based_direction_vector;
    angle_value        : angle_data_element;
    base_datum_plane   : datum_plane;
    direction_of_half_datum_plane : datum_based_direction_vector;
  INVERSE
    elements_of : SET [1:?] OF datum_based_vector_orientation FOR direction_components;
END_ENTITY; -- direction_element

( *
```

4.2.306 Documentation_layer_stratum

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Documentation_layer_stratum is a type of Stratum (see 4.2.845) used to describe patterns and other artwork information needed to manufacture a Stratum from materials which are not independently implementing a Physical_network (see 4.2.660) in the printed circuit board.

EXAMPLE 1 A Documentation_layer_stratum generally refers to those layers in domain terms for which there is a specifically designed shape other than, or in addition to, that of the printed circuit board edge, and for which the CAD systems traditionally output a separate master artwork pattern or file, but for which the resulting pattern is not for an independent conductive layer. For a "six layer pcb", in domain terms, there are six independently conductive design layers. This part of ISO 10303 allows the "outer layers" to be described by several Stratum to accommodate the different metallurgies that are found on electroless copper plated, electrolytic copper plated, and solder plated or solder coated Stratum on the primary and secondary sides of the Interconnect_module (see 4.2.440). The usual design view of the shape of these "outer layers" is that the shapes are identical for all practical purposes and the required shape is the cumulative shape that will appear after all the manufacturing processes are completed. A Stratum of conductive material, such as solder paste, is not an independently conductive layer in this context, since its shape is usually dependent upon the independently conductive layer over which it is applied through openings in solder mask or another Documentation_layer_stratum.

NOTE The general case of dielectric layers such as the adhesive Stratum, or the Stratum consisting of the dielectric substrate between two copper foil Stratum in a printed circuit board, is that they are neither a Documentation_layer_stratum nor a Design_layer_stratum. Their shapes are totally dependent upon the overall shape of the printed circuit board and its various passages.

EXAMPLE 2 If the stratum purpose is that of a coverlay, the insulating sheet on the outer surface of a flexible printed harness and the die-cut passages in the dielectric material of the coverlay are defined as Inter_stratum_features. A Stratum is used to describe the presence of the material. This Stratum in this context is neither a Design_layer_stratum (see 4.2.275) nor a Documentation_layer_stratum.

EXAMPLE 3 A Documentation_layer_stratum differs from a Design_layer_stratum in that its Ee_materials include such materials as solder maskant, epoxy ink, or dielectric materials; or materials such as solder paste that are described as conductive by the Ee_material (see 4.2.318) of the Stratum but whose shape is dependent upon that of an independently described Stratum.

EXAMPLE 4 An example of the shape of an independently described Stratum of conductive materials is that of the copper to be found in the traces and lands on the printed circuit board. The shape of a Stratum of solder paste is often dependent upon the shape of the Stratum of solder mask that in turn is dependent upon the shape of the circuitry on the primary and secondary sides of the Interconnect_module over which the solder mask and solder paste are applied.

The data associated with a Documentation_layer_stratum are the following:

— of_technology.

4.2.306.1 of_technology

The of_technology specifies the role of the Documentation_layer_technology (see 4.2.307) for the Documentation_layer_stratum.

NOTE EXPRESS specification:

```
*)
ENTITY documentation_layer_stratum
  SUBTYPE OF (stratum);
  SELF\stratum.of_technology : documentation_layer_technology;
END_ENTITY; -- documentation_layer_stratum
(*
```

4.2.307 Documentation_layer_technology

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Documentation_layer_technology is a type of Stratum_technology (see 4.2.871) that defines the purpose for the Documentation_layer_stratum (see 4.2.306).

The data associated with a Documentation_layer_technology are the following:

— documentation_layer_purpose.

4.2.307.1 documentation_layer_purpose

The documentation_layer_purpose specifies the string that describes the purpose for the Documentation_layer_technology.

EXAMPLE 1 Examples of these purposes are "Silkscreen", "Solder Mask", and "Solder Paste".

NOTE EXPRESS specification:

```
*)
ENTITY documentation_layer_technology
  SUBTYPE OF (stratum_technology);
  documentation_layer_purpose : STRING;
END_ENTITY; -- documentation_layer_technology

--tt tc1 wd9 modified
(*
```

4.2.308 Dose_equivalent_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Dose_equivalent_data_element is a

The data associated with a `Dose_equivalent_data_element` are the following:

— `additional_data`.

4.2.308.1 `additional_data`

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY dose_equivalent_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- dose_equivalent_data_element

( *
```

4.2.309 `Edge_based_2d_wireframe`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Edge_based_2d_wireframe` is a type of `Feature_shape_definition` (see 4.2.372). An `Edge_based_2d_wireframe` is described by a two-dimensional graph of edges and vertices. The graph may be disconnected. Within the graph the edges do not intersect except at their boundaries, otherwise known as vertices. The geometry associated with a vertex shall be consistent with the geometry associated with any of the edges of which the vertex forms a boundary.

NOTE EXPRESS specification:

```
*)
ENTITY edge_based_2d_wireframe
  SUBTYPE OF (planar_shape);
END_ENTITY; -- edge_based_2d_wireframe

( *
```

4.2.310 Edge_based_wireframe

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Edge_based_wireframe is a type of Bound_volume_shape (see 4.2.76). An Edge_based_wireframe is described by a three-dimensional graph of edges and vertices. The graph may be disconnected. Within the graph the edges do not intersect except at their boundaries, otherwise known as vertices. The geometry associated with a vertex shall be consistent with the geometry associated with any of the edges of which the vertex forms a boundary.

NOTE EXPRESS specification:

```
*)
ENTITY edge_based_wireframe
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- edge_based_wireframe

--tt tc1 wd9 added
(*
```

4.2.311 Edge_segment_cross_section

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Edge_segment_cross_section is a type of Volume_shape_intersection (see 4.2.960).

NOTE EXPRESS specification:

```

*)
ENTITY edge_segment_cross_section
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : edge_shape;
  SELF\volume_shape_intersection.projected_volume : part_template_3d_shape;
DERIVE
  cross_section_reference : edge_segment_vertex :=
  SELF\volume_shape_intersection.projected_volume\part_template_3d_shape.
  shape_characterized_part_template\inter_stratum_feature_edge_segment_template.
  start_vertex;

  associated_template : inter_stratum_feature_edge_segment_template :=
  SELF\volume_shape_intersection.projected_volume\part_template_3d_shape.
  shape_characterized_part_template;
INVERSE
  referencing_template : inter_stratum_feature_edge_segment_template FOR
  cross_section;
WHERE
  WR1: associated_template :=: referencing_template;
END_ENTITY; -- edge_segment_cross_section

--tt tc1 wd9 modified
( *

```

4.2.312 Edge_segment_vertex

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Edge_segment_vertex is a type of Managed_design_object (see 4.2.524). An Edge_segment_vertex supports explicit assignment of the order and connection of edge start and end points but without needing to use an explicit geometric or topological context.

The data associated with an Edge_segment_vertex are the following:

— name.

4.2.312.1 name

The name specifies the identifier for the Edge_segment_vertex. The name shall be unique within a population of Edge_segment_vertex.

Constraints:

UR1: The name shall be unique within a population of Edge_segment_vertex.

NOTE EXPRESS specification:

```

*)
ENTITY edge_segment_vertex
  SUBTYPE OF (single_datum,datum_point);
  UNIQUE
    UR1 : SELF\shape_element.name;
END_ENTITY; -- edge_segment_vertex

--tt tc1 wd9 modified
(

```

4.2.313 Edge_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Edge_shape is a type of Planar_shape (see 4.2.678) and a type of Feature_shape_definition (see 4.2.372). An Edge_shape may be one or more of a Bevel_edge_feature_shape (see 4.2.69), a Radius_edge_feature_shape (see 4.2.759), or a Step_edge_feature_shape (see 4.2.843). Edge_shape is provided to describe parametric data necessary to fabricate commonly used shapes on segments of interconnect substrates. Edge_shape applies to the entire arc length of the feature that is referencing it.

NOTE EXPRESS specification:

```

*)
ENTITY edge_shape
  ABSTRACT SUPERTYPE OF (ONEOF (
    bevel_edge_feature_shape,
    step_edge_feature_shape,
    radius_edge_feature_shape))
  SUBTYPE OF (planar_shape, feature_shape_definition);
(
  tt tc1 wd9 deleted
  local_path_datum : inter_stratum_feature_edge_segment_template;
*)
WHERE
  WR1: SIZEOF(QUERY ( escss &#60;* USEDIN (SELF,
    'AP210_ARM.VOLUME_SHAPE_INTERSECTION.SURFACE_INTERSECTION_RESULT')|
    NOT ('AP210_ARM.EDGE_SEGMENT_CROSS_SECTION' IN TYPEOF (escss))
  )) = 0;
END_ENTITY; -- edge_shape

(

```

4.2.314 Ee_approval

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_approval is a type of Managed_design_object (see 4.2.524). An Ee_approval is the dated indication that an Organization (see 4.2.569) or a Person_and_organization (see 4.2.647) has reviewed and approved an occurrence of something.

The data associated with an Ee_approval are the following:

- approval_status;
- authorizer;
- purpose;
- signoff_date.

4.2.314.1 approval_status

The approval_status specifies either approved, disapproved, not_yet_approved, or withdrawn as the state of consent applied to an aspect of product data as indicated by the Ee_approval.

4.2.314.2 authorizer

The authorizer specifies either a Person_and_organization (see 4.2.647) or an Organization (see 4.2.569) that authorizes the Ee_approval. There shall be one or more Person_and_organization or Organization for an Ee_approval.

4.2.314.3 purpose

The purpose specifies the text that provides the reason for the Ee_approval.

EXAMPLE 1 Typical values include "preliminary", "detailed", or "final".

4.2.314.4 signoff_date

The signoff_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) of the Ee_approval.

Constraints:

WR1: There shall be something to approve for approval to exist.

NOTE EXPRESS specification:

```
*)
ENTITY ee_approval
  SUBTYPE OF (managed_design_object);
  purpose      : text;
  authorizer   : SET [1:?] OF person_organization;
  approval_status : status;
  signoff_date : date_or_date_and_time;
END_ENTITY; -- ee_approval
(*
```

4.2.315 Ee_colour

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_colour is a type of Managed_design_object (see 4.2.524). An Ee_colour is the colour characterized by normalized quantities of the colours red, green, and blue in accordance with ISO/IEC 9592-1.

The data associated with an Ee_colour are the following:

— name.

4.2.315.1 name

The name specifies the string identifying the Ee_colour.

Constraints:

UR1: The name shall be unique within a population of Ee_colour.

NOTE EXPRESS specification:

```
*)
ENTITY ee_colour
  SUBTYPE OF (managed_design_object);
  name : ee_name;
UNIQUE
  UR1: name;
END_ENTITY; -- ee_colour
(*
```

4.2.316 Ee_document

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_document is a type of Managed_design_object (see 4.2.524). An Ee_document may be an Ee_specification (see 4.2.338). An Ee_document is a set of detailed information that describes a product or aspect of a product in detail. The information is in the form of text, tables, and graphics in a human readable language, although the hard copy may be a report from computer interpretable files.

The data associated with an Ee_document are the following:

- author;
- base_identifier;
- change_date;
- creation_date;
- document_approval;
- document_name;
- key_phrase;
- publisher;
- revision;
- security_code;
- text_usage.

4.2.316.1 author

The author specifies the Person (see 4.2.646) who creates an Ee_document. The author need not be specified for a particular Ee_document. There may be more than one Person for an Ee_document.

4.2.316.2 base_identifier

The base_identifier specifies the role of the Ee_document_identifier (see 4.2.317) for the Ee_document.

4.2.316.3 change_date

The change_date specifies the unique Date_and_time (see 4.2.213) when updates are made to an Ee_document. The change_date need not be specified for a particular Ee_document. There may be more than one Date_and_time for an Ee_document.

4.2.316.4 creation_date

The creation_date specifies the Date_and_time (see 4.2.213) when an Ee_document was created.

4.2.316.5 document_approval

The document_approval specifies the Ee_approval (see 4.2.314) of the Ee_document. The document_approval need not be specified for a particular Ee_document.

4.2.316.6 document_name

The document_name specifies the string identifying the Ee_document.

4.2.316.7 key_phrase

The key_phrase specifies the Structured_text (see 4.2.874) that details specific aspects within the text. The key_phrase need not be specified for a particular Ee_document. There may be more than one Structured_text for an Ee_document.

4.2.316.8 publisher

The publisher specifies the Organization (see 4.2.569) or the Person_and_organization (see 4.2.647) responsible for the Ee_document.

4.2.316.9 revision

The revision specifies the string identifying the configuration controlled version of the Ee_document.

NOTE If the originating organization does not configuration control documents, it is recommended to supply the value 'not used'.

4.2.316.10 security_code

The security_code specifies the Security_classification (see 4.2.804) of a particular Ee_document. The security_code need not be specified for a particular Ee_document.

4.2.316.11 text_usage

The text_usage specifies either the Ee_text (see 4.2.339) or the Structured_text (see 4.2.874) for an Ee_document. The text_usage need not be specified for a particular Ee_document. There may be more than one Ee_text or Structured_text for an Ee_document.

NOTE EXPRESS specification:

```

*)
ENTITY ee_document
  SUPERTYPE OF (ee_specification)
  SUBTYPE OF (managed_design_object);
    publisher      : person_organization;
    key_phrase     : OPTIONAL SET [1:?] OF structured_text;
    document_approval : OPTIONAL ee_approval;
    base_identifier : ee_document_identifier;
    change_date    : OPTIONAL SET [1:?] OF date_and_time;
    creation_date  : date_and_time;
    security_code  : OPTIONAL security_classification;
    text_usage     : OPTIONAL plain_or_structured_text;
    author         : OPTIONAL SET [1:?] OF person;
    revision       : ee_name;
    document_name  : ee_name;
UNIQUE
  UR1 : base_identifier, revision;
END_ENTITY; -- ee_document
( *

```

4.2.317 Ee_document_identifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_document_identifier is a type of Managed_design_object (see 4.2.524). An Ee_document_identifier is the combination of an identifier name and an assigned value.

EXAMPLE 1 An example of an Ee_document_identifier is an International Serial Book Number, ISBN, as the identifier_name, and the number associated with a specific publication as the assigned_value. Other examples may be found in the Library of Congress system and the Dewey Decimal System.

The data associated with an Ee_document_identifier are the following:

- assigned_value;
- identifier_name.

4.2.317.1 assigned_value

The assigned_value specifies a string identifying the content of the Ee_document_identifier.

4.2.317.2 identifier_name

The identifier_name specifies the string that establishes the context or type for the Ee_document_identifier.

Constraints:

UR1: The combination of the identifier_name and assigned_value shall be unique within a population of Ee_document_identifiers.

NOTE EXPRESS specification:

```
*)
ENTITY ee_document_identifier
  SUBTYPE OF (managed_design_object);
  identifier_name : ee_name;
  assigned_value  : ee_name;
UNIQUE
  UR1: identifier_name, assigned_value;
END_ENTITY; -- ee_document_identifier

( *
```

4.2.318 Ee_material

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_material is a type of Managed_design_object (see 4.2.524). An Ee_material is the tangible nature of a manufactured or natural product.

The data associated with an Ee_material are the following:

- material_name;
- registration_document;
- electrical_conductivity_classification;
- optical_conductivity_classification;
- thermal_conductivity_classification.

4.2.318.1 material_name

The material_name specifies the descriptive label that is registered for the Ee_material in the registration_document, and that represents the name of the Ee_material.

4.2.318.2 registration_document

The registration_document specifies a dictionary of registered materials in which the material_name is found.

4.2.318.3 electrical_conductivity_classification

The electrical_conductivity_classification specifies either conductive, non_conductive, resistive, semi_conductive. The electrical_conductivity_classification need not be specified for a particular Ee_material.

4.2.318.4 optical_conductivity_classification

The optical_conductivity_classification specifies either conductive, non_conductive, resistive, or semi_conductive. The optical_conductivity_classification need not be specified for a particular Ee_material.

4.2.318.5 thermal_conductivity_classification

The thermal_conductivity_classification specifies either conductive, non_conductive, resistive, or semi_conductive. The thermal_conductivity_classification need not be specified for a particular Ee_material.

Constraints:

UR1: The combination of material_name and registration_document shall be unique within a population of Ee_materials.

WR1: At least one of the following shall be provided: electrical_conductivity_classification, optical_conductivity_classification, thermal_conductivity_classification.

NOTE EXPRESS specification:

*)

```

ENTITY ee_material
  SUBTYPE OF (managed_design_object);
    material_name                : STRING;
-- tt tc1 wd8 deleted
-- material_description          : OPTIONAL nomenclature;
-- coordinated_material_property : OPTIONAL SET [1:?] OF
--                               coordinated_characteristic;
    electrical_conductivity_classification : OPTIONAL material_conductivity_classification
    optical_conductivity_classification   : OPTIONAL material_conductivity_classification
    registration_document                 : ee_document;
    thermal_conductivity_classification   : OPTIONAL material_conductivity_classification
UNIQUE
  UR1: material_name, registration_document;
END_ENTITY; -- ee_material

```

(*

4.2.319 Ee_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product is a type of Managed_design_object (see 4.2.524). An Ee_product is an item that is intended to be produced by or employed in a process. For the purposes of this part of ISO 10303, an Ee_product may be the result of a human endeavour that is not necessarily a manufacturing process. This standard predefines a small number of product classes for standardization purposes.

The data associated with an Ee_product are the following:

- design_owner;
- product_classification;
- product_nomenclature;
- part_number;
- product_version;
- standard_product_indicator.

4.2.319.1 design_owner

The design_owner specifies either the Organization (see 4.2.569) or the Person_and_organization (see 4.2.647) that is responsible for the Ee_product.

4.2.319.2 product_classification

The product_classification is an inverse relationship that specifies that the existence of the Ee_product is dependent on the existence of the Ee_product_related_product_category (see 4.2.326) that specifies the Ee_product as its of_product. The classifications specified by this Application Object are considered to be predefined in this standard. The predefined classifications are assembly, bare_die, functionality, interconnect, material, package, physical network design, piece part, printed part, requirements model, simulation model, standard part, technology specific model, and template model.

NOTE Those classifications may be related to specific types of Ee_product_definition that are technology specific, or for which there are no commonly understood domain term, or where there are many domain terms that may apply.

4.2.319.3 product_nomenclature

The product_nomenclature specifies the name by which the Ee_product is commonly known within an organization.

4.2.319.4 part_number

The part_number specifies the name used as the specific identification of an Ee_product for a particular organization.

4.2.319.5 product_version

The product_version is an inverse relationship that specifies that the existence of the Ee_product is dependent on the existence of the Ee_product_version (see 4.2.328) that specifies the Ee_product as its versioned_item. There shall be one or more Ee_product_version for an Ee_product.

4.2.319.6 standard_product_indicator

The standard_product_indicator specifies whether or not an Ee_product is considered to be a standard Product for a company, industry or other organization.

Constraints:

UR1: The combination of design_owner and part_number shall be unique within a population of Ee_products.

WR1: Each product that is referenced by a predefined product category shall be a product for the required subtype of Ee_product_definition.

EXAMPLE 1 If a product is classified as a package, there shall be an Ee_product_definition (see 4.2.323) that references the Ee_product through the version.versioned_item path.

NOTE EXPRESS specification:

*)

ENTITY ee_product

SUBTYPE OF (managed_design_object);

description : OPTIONAL text;

product_nomenclature : ee_name;

part_number : ee_name;

design_owner : person_organization;

standard_product_indicator : OPTIONAL BOOLEAN;

INVERSE

product_version : SET [1:?] OF ee_product_version FOR versioned_item;

product_classification : SET [1:?] OF ee_product_related_product_category FOR of_product;

UNIQUE

UR1 : design_owner, part_number;

WHERE

WR1 : assigned_product_category_correlation(product_version, product_classification);

END_ENTITY; -- ee_product

(*

4.2.320 Ee_product_category

An Ee_product_category is a type of Managed_design_object (see 4.2.524). An Ee_product_category is a class in a class hierarchy for the purpose of categorizing products according to their characteristics. An Ee_product_category may be an Ee_product_related_product_category (see 4.2.326).

The data associated with an Ee_product_category are the following:

- category_parameter;
- category_specific_values;
- description;
- name;
- reference_document.

4.2.320.1 category_parameter

The category_parameter specifies the Model_parameter (see 4.2.549) assigned to this Ee_product_category for categorization purposes. All sub-categories of this Ee_product_category shall inherit the category_parameter. There shall be one or more Model_parameter for an Ee_product_category.

4.2.320.2 category_specific_values

The category_specific_values specifies the Parameter_assignment (see 4.2.603) that assigns a specific value to a specific Model_parameter (see 4.2.549) for this Ee_product_category. All sub-categories of this Ee_product_category shall inherit the category_specific_values. There shall be one or more Parameter_assignment for an Ee_product_category.

4.2.320.3 description

The description specifies one role of a string for the Ee_product_category.

4.2.320.4 name

The name specifies the string for the identifier for the Ee_product_category.

4.2.320.5 reference_document

The reference_document specifies the Ee_document (see 4.2.316) that is the reference for the Ee_product_category. The reference_document establishes the scope for the name. All Ee_product_category names defined in this standard shall use this standard as the reference_document.

NOTE The use of reference_document allows organizations to resolve name clashes between Ee_product_category instances.

EXAMPLE 1 This standard pre-defines the name 'assembly' while an organization might use 'PWA' for that purpose. The inclusion of the reference_document permits both to be in a product category tree and pointing to the same products. An application conformant to this standard looking for pre-defined names would use the reference_document as a filter and ignore 'PWA'.

Constraints:

UR1: The name in combination with the reference_document shall be unique within a population of Ee_product_categories within an enterprise.

WR1: Each category_specific_value shall be assigned to one member of Model_parameter that is in the role of category_parameter in a higher level category of this Ee_product_category.

WR2: All Ee_product_category names defined in this standard shall use this standard as the reference_document.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_category
  SUPERTYPE OF (ee_product_related_product_category)
  SUBTYPE OF (managed_design_object);
  category_parameter      : OPTIONAL SET [1:?] OF model_parameter;
  category_specific_values : OPTIONAL SET [1:?] OF parameter_assignment;
  description              : OPTIONAL ee_name;
  name                    : ee_name;
  reference_document       : ee_document;
UNIQUE
  UR1: name, reference_document;
END_ENTITY; -- ee_product_category

( *
```

4.2.321 Ee_product_category_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_category_relationship is a type of Managed_design_object (see 4.2.524). An Ee_product_category_relationship establishes an inheritance relationship that is a parent-child relationship between two classes in a class hierarchy of Ee_product_category (see 4.2.320).

The data associated with an Ee_product_category_relationship are the following:

- category;
- sub_category.

4.2.321.1 category

The category specifies the Ee_product_category (see 4.2.320) that is in the role of higher level category in the classification structure.

4.2.321.2 sub_category

The sub_category specifies the Ee_product_category (see 4.2.320) that inherits properties from the higher level category in the classification structure.

Constraints:

WR1: The Ee_product_category_relationship shall not establish a cyclic path in the classification structure it is a member of.

NOTE EXPRESS specification:

```
*)  
ENTITY ee_product_category_relationship  
  SUBTYPE OF (managed_design_object);  
    sub_category : ee_product_category;  
    category      : ee_product_category;  
END_ENTITY; -- ee_product_category_relationship  
  
(*
```

4.2.322 Ee_product_configuration

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_configuration is a type of Managed_design_object (see 4.2.524). An Ee_product_configuration is the association of a product version with a model name that enterprise marketing provides to a customer.

The data associated with an Ee_product_configuration are the following:

- configuration_manager;
- item_id;
- model_of_product;
- phase_of_product;
- product_configuration_approval;
- satisfying_product.

4.2.322.1 configuration_manager

The configuration_manager specifies either the Organization (see 4.2.569) or the Person_and_organization (see 4.2.647) that manages the Ee_product_configuration.

4.2.322.2 item_id

The item_id specifies the identification of an Ee_product_configuration.

4.2.322.3 model_of_product

The model_of_product specifies the role of the Ee_product_model (see 4.2.325) for the Ee_product_configuration.

4.2.322.4 phase_of_product

The phase_of_product specifies the string that defines the life cycle phase of the Ee_product_configuration.

EXAMPLE 1 Examples of phase_of_product include "Preliminary design", "Prototype", "Released for Production", "Obsolete for New Manufacture", and "Retired".

4.2.322.5 product_configuration_approval

The product_configuration_approval specifies the Ee_approval (see 4.2.314) that is associated with the Ee_product_configuration.

4.2.322.6 satisfying_product

The satisfying_product specifies the Ee_product_version (see 4.2.328) that has been selected to satisfy a real or perceived customer need for the Ee_product_configuration. The satisfying_product need not be specified for a particular Ee_product_configuration. There may be more than one Ee_product_version for an Ee_product_configuration.

NOTE In some enterprises the marketing department selects product model nomenclature independent of the existence of actual product versions to satisfy the marketing requirements. This object permits the enterprise to explicitly manage the relationship between model nomenclature and the product version.

Constraints:

UR1: The combination of item_id and satisfying_product shall be unique within a population of Ee_product_configurations.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_configuration
  SUBTYPE OF (managed_design_object);
    satisfying_product      : OPTIONAL SET [1:?] OF ee_product_version;
    configuration_manager   : person_organization;
    item_id                 : STRING;
    product_configuration_approval : ee_approval;
    model_of_product        : ee_product_model;
    phase_of_product        : STRING;
  UNIQUE
    UR1 : item_id, satisfying_product;
END_ENTITY; -- ee_product_configuration

--tt tc1 wd9 modified
( *
```

4.2.323 Ee_product_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_definition is a type of Managed_design_object (see 4.2.524). An Ee_product_definition is either a Functional_unit_definition (see 4.2.401) or a Physical_unit (see 4.2.664) or additionally may be an Externally_defined_ee_product_definition (see 4.2.364) or additionally may be a Library_ee_product_definition (see 4.2.513).

The data associated with an Ee_product_definition are the following:

- cad_filename;
- context_description;
- creation_date;
- creator;
- discipline_id;
- documentation;
- product_definition_approval;
- product_definition_id;
- security_code;
- version.

NOTE This entity may be used to capture the definition of a particular Ee_product_version at any intermediate stage of its development where the definition is not formally tracked by an organization at the Ee_product_version level. It may be used to capture the various stages in the definition cycle of a product.

4.2.323.1 cad_filename

The cad_filename specifies the name of the file or data repository that contains the detailed Ee_product_definition in a computer aided design system. The reference is interpreted to be to an external definition of the details of the Ee_product_definition that is meaningful only within the organization that designed the product. The cad_filename need not be specified for a particular Ee_product_definition.

NOTE The data contained in the cad_filename may include more information to allow the enterprise to identify the originating system in addition to a simple name.

EXAMPLE 1 As an example, the following information may be included: operating system supplier name = "xxx", operating system name = "xxx", operating system version = "xxx", cad tool supplier name = "xxx", cad tool name = "xxx", cad tool version = "xxx", cad filename = "xxx", cad filename date stamp = "xxx" where "xxx" is the value information.

4.2.323.2 context_description

The context_description specifies the Ee_text (see 4.2.339) that is the purpose for a particular definition of the Ee_product_definition. The context_description is optional and need not be supplied.

4.2.323.3 creation_date

The creation_date specifies the Date_and_time (see 4.2.213) at which the Ee_product_definition was first defined.

4.2.323.4 creator

The creator specifies the Person_and_organization (see 4.2.647) or the Organization (see 4.2.569) that initially created the Ee_product_definition. There shall be one or more Person_and_organization or Organization for an Ee_product_definition.

4.2.323.5 discipline_id

The discipline_id specifies the name of the discipline that creates the Ee_product_definition.

EXAMPLE 1 Typical usage may include one of the following as the last item in a comma separated list: analysis, synthesis.

4.2.323.6 documentation

The documentation specifies the role of the Ee_document (see 4.2.316) for the Ee_product_definition. The documentation need not be specified for a particular Ee_product_definition. There may be more than one Ee_document for an Ee_product_definition.

4.2.323.7 product_definition_approval

The product_definition_approval specifies the role of an Ee_approval (see 4.2.314) for the Ee_product_definition.

4.2.323.8 product_definition_id

The product_definition_id specifies the name used for identification of the Ee_product_definition so that users of the information can reference it.

4.2.323.9 security_code

The security_code specifies the role of the Security_classification (see 4.2.804) for the Ee_product_definition. The security_code is optional and need not be supplied for the Ee_product_definition.

4.2.323.10 version

The version specifies the role of the Ee_product_version (see 4.2.328) for the Ee_product_definition.

Constraints:

UR1: The combination of the product_definition_id and the version shall be unique within a population of Ee_product_definitions.

UR2: The combination of the discipline_id and the version shall be unique within a population of Ee_product_definitions.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_definition
  ABSTRACT SUPERTYPE OF (ONEOF (
    analytical_model_definition,
    ee_rule_definition,
    evaluation_ee_product_definition,
    footprint_definition,
    functional_unit_definition,
    material_definition,
    padstack_definition,
    physical_unit,
    physical_unit_network_definition,
    requirement_definition,
    stratum,
    template_definition) ANDOR
    externally_defined_ee_product_definition)
  SUBTYPE OF (managed_design_object);
  creator                : SET [1:?] OF person_organization;
  product_definition_id  : ee_name;
  security_code          : OPTIONAL security_classification;
  context_description    : OPTIONAL ee_text;
  version                : ee_product_version;
  product_definition_approval : ee_approval;
  creation_date          : date_and_time;
  documentation          : OPTIONAL SET [1:?] OF ee_document;
  cad_filename           : OPTIONAL ee_name;
  discipline_id          : OPTIONAL ee_text;
  life_cycle_status      : text;
  UNIQUE
    UR1 : product_definition_id, version;
END_ENTITY; -- ee_product_definition
(

```

4.2.324 Ee_product_definition_based_constraint_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Ee_product_definition_based_constraint_occurrence` is a type of `Constraint_occurrence` (see 4.2.193) that has a basis in the characteristics of an `Ee_product_definition` (see 4.2.323) and that identifies the basis definition. An `Ee_product_definition_based_constraint_occurrence` may be a `Shape_and_ee_product_definition_based_constraint_occurrence` (see 4.2.806).

The data associated with an `Ee_product_definition_based_constraint_occurrence` are the following:

— `constraining_definition`.

4.2.324.1 **constraining_definition**

The `constraining_definition` specifies the role of the `Ee_product_definition` (see 4.2.323) for the `Ee_product_definition_based_constraint_occurrence`.

NOTE EXPRESS specification:

```
*)
ENTITY ee_product_definition_based_constraint_occurrence
  SUPERTYPE OF (shape_and_ee_product_definition_based_constraint_occurrence)
  SUBTYPE OF (constraint_occurrence);
  constraining_part : ee_product_definition;
END_ENTITY; -- ee_product_definition_based_constraint_occurrence
```

(*

4.2.325 **Ee_product_model**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Ee_product_model` is a type of `Managed_design_object` (see 4.2.524). An `Ee_product_model` is the nomenclature that an enterprise marketing function assigns for purposes of communication with customers.

The data associated with an `Ee_product_model` are the following:

— `model_name`.

4.2.325.1 **model_name**

The `model_name` specifies the `Ee_text` (see 4.2.339) identifier of the `Ee_product_model`.

Constraints:

UR1: The `model_name` shall be unique within a population of `Ee_product_models`.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_model
  SUBTYPE OF (managed_design_object);
  model_name : ee_text;
  UNIQUE
  UR1 : model_name;
END_ENTITY; -- ee_product_model

( *

```

4.2.326 Ee_product_related_product_category

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_related_product_category is a type of Ee_product_category (see 4.2.320). An Ee_product_related_product_category relates one or more products to the specific characteristics associated with the category.

The data associated with an Ee_product_related_product_category are the following:

— of_product.

4.2.326.1 of_product

The of_product specifies the Ee_product (see 4.2.319) assigned to this Ee_product_related_product_category. There shall be one or more Ee_product for an Ee_product_related_product_category.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_related_product_category
  SUBTYPE OF (ee_product_category);
  of_product : SET [1:?] OF ee_product;
END_ENTITY; -- ee_product_related_product_category

( *

```

4.2.327 Ee_product_specific_parameter_value_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_specific_parameter_value_assignment is a type of Managed_design_object (see 4.2.524). An Ee_product_specific_parameter_value_assignment is the assignment of a specific value to a specific characteristic of an Ee_product_category for a specific product.

The data associated with an Ee_product_specific_parameter_value_assignment are the following:

- assigned_parameter;
- associated_product_category;
- of_product.

4.2.327.1 assigned_parameter

The assigned_parameter specifies the Parameter_assignment (see 4.2.603) that is assigning the specific value for this Ee_product_specific_parameter_value_assignment.

4.2.327.2 associated_product_category

The associated_product_category specifies the Ee_product_related_product_category (see 4.2.326) that includes the characteristic being assigned a value with this Ee_product_specific_parameter_value_assignment.

4.2.327.3 of_product

The of_product specifies the Ee_product (see 4.2.319) assigned a characteristic value by this Ee_product_specific_parameter_value_assignment.

Constraints:

WR1: Each Model_parameter in the role of assigned_parameter.assigned_parameter shall be in the role of category_parameter for the Ee_product_related_product_category specified by associated_product_category.

WR2: Each Ee_product in the role of of_product shall be in the role of of_product for the Ee_product_related_product_category specified by associated_product_category.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_specific_parameter_value_assignment
  SUBTYPE OF (managed_design_object);
    associated_product_category : ee_product_related_product_category;
    of_product                  : ee_product;
    assigned_parameter          : parameter_assignment;
END_ENTITY; -- ee_product_specific_parameter_value_assignment

( *
```

4.2.328 Ee_product_version

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_product_version is a type of Managed_design_object (see 4.2.524). An Ee_product_version is the identification of the representation of a product after its design has reached a point of control or has undergone a formal release or change.

The data associated with an Ee_product_version are the following:

- authorizing_agreement;
- creator;
- product_definition;
- product_version_approval;
- revision_code;
- security_code;
- versioned_item.

4.2.328.1 authorizing_agreement

The authorizing_agreement specifies the role of the Contract (see 4.2.196) for the Ee_product_version. The authorizing_agreement need not be provided for a specific Ee_product_version.

4.2.328.2 creator

The creator specifies the Organization (see 4.2.569) or the Person_and_organization (see 4.2.647) responsible for the Ee_product_version.

4.2.328.3 product_definition

The product_definition is an inverse relationship that specifies that the existence of the Ee_product_version is dependent on the existence of the Ee_product_definition (see 4.2.323) that specifies the Ee_product_version as its version. There shall be one or more Ee_product_definition for an Ee_product_version.

4.2.328.4 product_version_approval

The product_version_approval specifies the Ee_approval (see 4.2.314) associated with the Ee_product_version.

4.2.328.5 revision_code

The revision_code specifies the string identifying an Ee_product_version.

4.2.328.6 security_code

The security_code specifies the Security_classification (see 4.2.804) of an Ee_product_version and is used to control access to design and other information about the Ee_product_version. The security_code is optional and need not be supplied.

4.2.328.7 versioned_item

The versioned_item specifies the Ee_product (see 4.2.319) of which the Ee_product_version identifies the version.

Constraints:

UR1: The combination of the versioned_item and the revision_code shall be unique within a population of Ee_product_versions.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_version
  SUPERTYPE OF (ee_rule_version)
  SUBTYPE OF (managed_design_object);
  creator                : person_organization;
  authorizing_agreement  : OPTIONAL contract;
  revision_code          : ee_name;
  versioned_item         : ee_product;
  security_code          : OPTIONAL security_classification;
  product_version_approval : ee_approval;
  INVERSE
    product_definition : SET [1:?] OF ee_product_definition FOR version;
  UNIQUE
    UR1 : versioned_item, revision_code;
END_ENTITY; -- ee_product_version
( *
```

4.2.329 Ee_product_version_supersedence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Ee_product_version_supersedence is a type of Managed_design_object (see 4.2.524).

The data associated with an Ee_product_version_supersedence are the following:

- superseded_product_version;
- superseding_product_version.

4.2.329.1 superseded_product_version

The superseded_product_version specifies an Ee_product_version (see 4.2.328) for the Ee_product_version_supersedence.

4.2.329.2 superseding_product_version

The superseding_product_version specifies an Ee_product_version (see 4.2.328) for the Ee_product_version_supersedence.

Constraints:

UR1: The combination of superseded_product_version and superseding_product_version shall be unique within a population of Ee_product_version_supersedence.

NOTE EXPRESS specification:

```

*)
ENTITY ee_product_version_supersedence
  SUBTYPE OF (managed_design_object);
    superseded_product_version : ee_product_version;
    superseding_product_version : ee_product_version;
  UNIQUE
    UR1 : superseded_product_version, superseding_product_version;
END_ENTITY; -- ee_product_version_supersedence

( *
```

4.2.330 Ee_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Ee_requirement_occurrence` is a type of `Managed_design_object` (see 4.2.524). An `Ee_requirement_occurrence` may be a `Requirement_group_occurrence` (see 4.2.772), and additionally may be one of a `Constraint_occurrence` (see 4.2.193), a `Design_characteristic_occurrence` (see 4.2.261), a `Design_requirement_occurrence` (see 4.2.279), a `Inspection_requirement_occurrence` (see 4.2.430), a `Test_requirement_occurrence` (see 4.2.896), or a `Purpose_occurrence` (see 4.2.757) and additionally may be one of an `Electrical_requirement_occurrence` (see 4.2.346), an `Mechanical_requirement_occurrence` (see 4.2.544), an `Electromagnetic_requirement_occurrence` (see 4.2.348) or a `Thermal_requirement_occurrence` (see 4.2.908), and additionally may be one of an `Item_restricted_requirement_occurrence` (see 4.2.478) or a `Layout_spacing_requirement_occurrence` (see 4.2.509). An `Ee_requirement_occurrence` is a statement of criteria for design or manufacturing that may be derived from more abstract criteria or may be a decomposition of a technical parameter associated with a product structure. Complex requirements may be associated with a product specifically for the purpose of managing the requirements.

NOTE This part of ISO 10303 provides generic categories for enterprise classification of requirements and includes some domain classified requirements.

EXAMPLE 1 An inspection requirement may be derived from an electromagnetic requirement.

The data associated with an `Ee_requirement_occurrence` are the following:

- `required_analytical_representation`;
- `required_functional_specification`;
- `description`;
- `name`;
- `hierarchy_context`;
- `required_characteristic`;
- `required_coordinated_characteristic`;
- `required_material`;
- `required_part`;
- `requirement_specification`.

4.2.330.1 `required_analytical_representation`

The `required_analytical_representation` specifies the role of the `Analytical_representation` for the `Ee_requirement_occurrence`. There may be more than one `Analytical_representation` for an `Ee_requirement_occurrence`. The `required_analytical_representation` need not be specified for a particular `Ee_requirement_occurrence`.

4.2.330.2 required_functional_specification

The `required_functional_specification` specifies the role of the `Functional_specification` for the `Ee_requirement_occurrence`. There may be more than one `Functional_specification` for an `Ee_requirement_occurrence`. The `required_functional_specification` need not be specified for a particular `Ee_requirement_occurrence`.

4.2.330.3 description

The `description` specifies the `Ee_text` (see 4.2.339) of details of the `Ee_requirement_occurrence` in a natural language. The `description` need not be specified for a particular `Ee_requirement_occurrence`.

4.2.330.4 name

The `name` specifies the unique string that identifies the `Ee_requirement_occurrence`.

NOTE The `name` should include information identifying the source of the requirement.

4.2.330.5 hierarchy_context

The `hierarchy_context` is an inverse relationship that specifies the `Requirement_composition` (see 4.2.770) that specifies the `Ee_requirement_occurrence` as its element. There shall be no more than one `Requirement_composition` for an `Ee_requirement_occurrence`.

4.2.330.6 required_characteristic

The `required_characteristic` specifies the role of the `Characteristic` (see 4.2.98) for the `Ee_requirement_occurrence`. The `required_characteristic` need not be specified for a particular `Ee_requirement_occurrence`. There may be more than one `Characteristic` for an `Ee_requirement_occurrence`.

4.2.330.7 required_coordinated_characteristic

The `required_coordinated_characteristic` specifies the role of the `Coordinated_characteristic` (see 4.2.197) for an `Ee_requirement_occurrence`. The `required_coordinated_characteristic` need not be specified for a particular `Ee_requirement_occurrence`. There may be more than one `Coordinated_characteristic` for an `Ee_requirement_occurrence`.

4.2.330.8 required_material

The `required_material` specifies the role of the `Ee_material` (see 4.2.318) for the `Ee_requirement_occurrence`. The `required_material` need not be specified for a particular `Ee_requirement_occurrence`.

4.2.330.9 required_part

The required_part specifies the role of the Ee_product (see 4.2.319) for the Ee_requirement_occurrence. The required_part need not be specified for a particular Ee_requirement_occurrence. There may be more than one Ee_product for an Ee_requirement_occurrence.

4.2.330.10 requirement_specification

The requirement_specification specifies the Ee_specification (see 4.2.338) for the source of the Ee_requirement_occurrence. There shall be one or more Ee_specification for an Ee_requirement_occurrence.

Constraints:

UR1: The associated_definition shall be unique within the population of Ee_requirement_occurrence.

NOTE EXPRESS specification:

```

*)
ENTITY ee_requirement_occurrence
  SUPERTYPE OF (((ONEOF (thermal_requirement_occurrence,
    electromagnetic_requirement_occurrence,
    electrical_requirement_occurrence,
    mechanical_requirement_occurrence) ANDOR ONEOF (
    design_requirement_occurrence,
    inspection_requirement_occurrence,
    test_requirement_occurrence,
    purpose_occurrence,
    constraint_occurrence,
    design_characteristic_occurrence)) ANDOR
    ONEOF (layout_spacing_requirement_occurrence,
    item_restricted_requirement_occurrence)) ANDOR
    requirement_group_occurrence)
  SUBTYPE OF (managed_design_object);
  associated_definition          : requirement_definition;
  required_analytical_representation : OPTIONAL SET [1:?] OF
    analytical_representation;
  required_functional_specification : OPTIONAL SET [1:?] OF
    functional_specification;
  required_part                  : OPTIONAL SET [1:?] OF ee_product;
  required_coordinated_characteristic : OPTIONAL SET [1:?] OF
    coordinated_characteristic;
  requirement_specification      : SET [1:?] OF ee_specification;
  required_characteristic        : OPTIONAL SET [1:?] OF characteristic;
  required_material              : OPTIONAL ee_material;
  description                    : OPTIONAL ee_text;
  name                          : ee_name;
  INVERSE
    hierarchy_context : SET [0:1] OF requirement_composition FOR element;
  UNIQUE
    UR1: associated_definition;
END_ENTITY; -- ee_requirement_occurrence
( *

```

4.2.331 Ee_rule_complex_clause

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

The Ee_rule_complex_clause is a type of Managed_design_object (see 4.2.524). An Ee_rule_complex_clause consist of one or more simple clauses that are ANDed together. The AND operator is provided by the SET aggregation in this application object.

The data associated with an Ee_rule_complex_clause are the following:

- conjunctive_content;
- identifier.

4.2.331.1 conjunctive_content

The conjunctive_content specifies the Ee_rule_simple_clause (see 4.2.336) for the Ee_rule_complex_clause. There shall be one or more conjunctive_content specified for an Ee_rule_complex_clause.

4.2.331.2 identifier

The identifier specifies the name for the Ee_rule_complex_clause.

Constraints:

UR1: The identifier shall be unique within a population of Ee_rule_complex_clause.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_complex_clause
  SUBTYPE OF ( managed_design_object );
    conjunctive_content : SET [1:?] OF ee_rule_simple_clause;
    identifier           : STRING ;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- ee_rule_complex_clause

( *
```

4.2.332 Ee_rule_conclusion_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_rule_conclusion_definition is a type of Managed_design_object (see 4.2.524). An Ee_rule_conclusion_definition provides the conclusion part of a rule construct.

The data associated with an Ee_rule_conclusion_definition are the following:

- rule_definition;
- identifier;
- conjunctive_content.

4.2.332.1 rule_definition

The rule_definition specifies the Ee_rule_definition (see 4.2.333) for the Ee_rule_conclusion_definition. The rule_definition is the rule being composed by this Ee_rule_conclusion_definition.

4.2.332.2 identifier

The identifier specifies the name for the Ee_rule_conclusion_definition.

4.2.332.3 conjunctive_content

The conjunctive_content specifies specifies the Ee_rule_simple_clause for the Ee_rule_conclusion_definition. There shall be one or more conjunctive_content specified for an Ee_rule_conclusion_definition.

Constraints:

UR1: The combination of the rule_definition and the identifier shall be unique within a population of Ee_rule_conclusion_definition.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_conclusion_definition
  SUBTYPE OF ( managed_design_object );
    rule_definition      : ee_rule_definition;
    identifier           : STRING ;
    conjunctive_content  : SET [1:?] OF ee_rule_simple_clause;
  UNIQUE
    UR1 : rule_definition, identifier;
END_ENTITY; -- ee_rule_conclusion_definition

( *
```

4.2.333 Ee_rule_definition

An Ee_rule_definition is a type of Ee_product_definition (see 4.2.323). An Ee_rule_definition is a formal specification that defines a constraint by relating two elements based upon characteristics of a design, one as premise and one as conclusion. The Ee_rule_definition is for one version of a rule only.

The data associated with an Ee_rule_definition are the following:

- conclusion;
- defined_versioned_rule;
- premise.

4.2.333.1 conclusion

The conclusion specifies the role of the ee_rule_conclusion_definition for the Ee_rule_definition.

4.2.333.2 defined_versioned_rule

The version specifies the role of the Ee_rule_version (see 4.2.337) for the Ee_rule_definition.

4.2.333.3 premise

The premise specifies the role of the ee_rule_premise_definition for the Ee_rule_definition.

NOTE EXPRESS specification:

```
*)
ENTITY ee_rule_definition
  SUBTYPE OF (ee_product_definition);
  SELF\ee_product_definition.version : ee_rule_version;
  INVERSE
    premise      : ee_rule_premise_definition FOR rule_definition;
    conclusion   : ee_rule_conclusion_definition FOR rule_definition;
END_ENTITY; -- ee_rule_definition
(*
```

4.2.334 Ee_rule_function

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_rule_function is a type of Managed_design_object (see 4.2.524). An Ee_rule_function is a formal specification for a process that includes the source code and instance values for the process. An Ee_rule_function has a unique combination of identifier and reference_document. An Ee_rule_function shall have at least one parameter.

NOTE Source code is the computer-interpretable expressions that form a part of the Ee_rule_function. The attribute source_code is the relationship of these expressions to the Ee_rule_function.

The data associated with an Ee_rule_function are the following:

- identifier;
- parameter;
- reference_document;
- source_code.

4.2.334.1 identifier

The identifier specifies the name for the Ee_rule_function.

4.2.334.2 parameter

The parameter specifies the role of the Model_parameter (see 4.2.549) for an Ee_rule_function. There shall be one or more Model_parameter specified for a particular Ee_rule_function.

4.2.334.3 reference_document

The reference_document specifies the role of the Ee_document (see 4.2.316) for an Ee_rule_function.

4.2.334.4 source_code

The source_code specifies the role of the Ee_specification (see 4.2.338) for an Ee_rule_function.

Constraints:

UR1: The combination of the identifier and the reference_document shall be unique within a population of Ee_rule_functions.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_function
  SUBTYPE OF (managed_design_object);
  identifier      : STRING;
  parameter       : SET [1:?] OF model_parameter;
  reference_document : ee_document;
  source_code     : ee_specification;
  UNIQUE
    UR1 : identifier, reference_document;
END_ENTITY; -- ee_rule_function

```

(*

4.2.335 Ee_rule_premise_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

The Ee_rule_premise_definition is a type of Managed_design_object (see 4.2.524). An Ee_rule_premise_definition is the premise for a rule construct. The Ee_rule_premise_definition uses an 'OR' construct so that if any of the parts of the premise are true, the rule will fire.

The data associated with an Ee_rule_premise_definition are the following:

- rule_definition;
- identifier;
- disjunctive_content.

4.2.335.1 rule_definition

The rule_definition specifies the Ee_rule_definition (see 4.2.333) for the Ee_rule_premise_definition. The rule_definition is the Ee_rule_definition this Ee_rule_premise_definition is helping to compose.

4.2.335.2 identifier

The identifier specifies the name for the Ee_rule_premise_definition.

4.2.335.3 disjunctive_content

The disjunctive_content specifies an Ee_rule_complex_clause (see 4.2.331). There shall be one or more disjunctive_content specified for an Ee_rule_premise_definition.

Constraints:

UR1: The combination of rule_definition and identifier shall be unique within a population of Ee_rule_- conclusion_definition.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_premise_definition
  SUBTYPE OF ( managed_design_object );
    rule_definition      : ee_rule_definition;
    identifier           : STRING ;
    disjunctive_content  : SET [1:?] OF ee_rule_complex_clause;
  UNIQUE
    UR1 : rule_definition, identifier;
END_ENTITY; -- ee_rule_premise_definition

```

(*

4.2.336 Ee_rule_simple_clause

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

The Ee_rule_simple_clause is a type of Managed_design_object (see 4.2.524). An Ee_rule_simple_clause assigns data specific to the rule definition to the generic algorithm specified by rule_function.

NOTE This data is not product instance data, but is categorical data helping to compose this definition.

EXAMPLE 1 An example of this type of data would be "all tracks less than 5 mils."

The data associated with an Ee_rule_simple_clause are the following:

- rule_function;
- parameters;
- identifier.

4.2.336.1 rule_function

The rule_function specifies the Ee_rule_definition (see 4.2.333) of the Ee_rule_simple_clause. The rule_function is the rule to which the Ee_simple_clause is assigning parameters.

4.2.336.2 parameters

The parameters specifies the Parameter_assignment (see 4.2.603) for an Ee_rule_simple_clause. There shall be one or more parameters specified for an Ee_rule_simple_clause.

4.2.336.3 identifier

The identifier specifies the name for the Ee_rule_simple_clause.

Constraints:

UR1: The identifier shall be unique within a population of Ee_rule_simple_clause.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_simple_clause
  SUBTYPE OF ( managed_design_object );
  rule_function : ee_rule_function;
  parameter      : SET [1:?] OF parameter_assignment;
  identifier      : STRING ;
  UNIQUE
    UR1: identifier;
END_ENTITY; -- ee_rule_simple_clause

( *
```

4.2.337 Ee_rule_version

An Ee_rule_version is a type of Ee_product_version (see 4.2.328). An Ee_rule_version is the management of the assignment of Ee_rule_definition (see 4.2.333) to Ee_rule (see ??). An Ee_rule may evolve with usage for any number of reasons.

The data associated with an Ee_rule_version are the following:

— management_action.

4.2.337.1 management_action

The management_action is an inverse relationship that specifies that the existence of the Ee_rule_version is dependent on the existence of the Rule_action (see 4.2.780) that specifies the Ee_rule_version as its subject_rule. There shall be one or more Rule_action for an Ee_rule_version.

NOTE EXPRESS specification:

```

*)
ENTITY ee_rule_version
  SUBTYPE OF (ee_product_version);
  INVERSE
    management_action : SET [1:?] OF rule_action FOR subject_rule;
END_ENTITY; -- ee_rule_version
( *

```

4.2.338 Ee_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_specification is a type of Ee_document (see 4.2.316). An Ee_specification may be either a Design_specification (see 4.2.280), a Language_reference_manual (see 4.2.497), a Material_specification (see 4.2.537), a Process_specification (see 4.2.740), or a Test_specification (see 4.2.898).

The data associated with an Ee_specification are the following:

— source.

4.2.338.1 source

The source specifies the Organization (see 4.2.569) or the Person_and_organization (see 4.2.647) responsible for the information of an Ee_specification.

NOTE EXPRESS specification:

```

*)
ENTITY ee_specification
  SUPERTYPE OF (ONEOF (design_specification, process_specification,
    material_specification, test_specification, language_reference_manual))
  SUBTYPE OF (ee_document);
  source : person_organization;
END_ENTITY; -- ee_specification
( *

```

4.2.339 Ee_text

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Ee_text is a type of Managed_design_object (see 4.2.524). An Ee_text is a string used to convey information.

The data associated with an Ee_text are the following:

— content.

4.2.339.1 content

The content specifies the organized list of one or more words that comprise a message.

NOTE EXPRESS specification:

```
*)  
ENTITY ee_text  
  SUBTYPE OF (managed_design_object);  
  content : STRING;  
END_ENTITY; -- ee_text
```

(*

4.2.340 Electric_charge_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Electric_charge_data_element is a

The data associated with an Electric_charge_data_element are the following:

— additional_data.

4.2.340.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY electric_charge_data_element  
  SUBTYPE OF (data_element);  
END_ENTITY; -- electric_charge_data_element
```

(*

4.2.341 Electric_current_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Electric_current_data_element is a

The data associated with an Electric_current_data_element are the following:

— additional_data.

4.2.341.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY electric_current_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- electric_current_data_element

( *
```

4.2.342 Electrical_isolation_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Electrical_isolation_laminate_component is a type of Filled_area_material_removal_laminate_component (see 4.2.384) that indicates that material is to be removed in the realization of the feature that is in the role of design_intent inherited from Filled_area_material_removal_laminate_component. The requirement shall be interpreted mathematically as a requirement for application of the regularized Boolean subtraction operation.

The data associated with an Electrical_isolation_laminate_component are the following:

— usage_definition.

4.2.342.1 usage_definition

The usage_definition specifies the role of the Electrical_isolation_removal_template (see 4.2.344) for the Electrical_isolation_laminate_component.

NOTE EXPRESS specification:

```

*)
ENTITY electrical_isolation_laminate_component
  SUBTYPE OF (filled_area_material_removal_laminate_component);
  SELF\assembly_component.usage_definition : electrical_isolation_removal_template;
END_ENTITY; -- electrical_isolation_laminate_component

( *

```

4.2.343 Electrical_isolation_removal_structure

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Electrical_isolation_removal_structure is a type of Filled_area_material_removal_structure (see 4.2.385) that has specific materials removed from specific subsets of the filled area for the purpose of electrical isolation.

The data associated with an Electrical_isolation_removal_structure are the following:

— electrical_isolation_spacing_requirement.

4.2.343.1 electrical_isolation_spacing_requirement

The electrical_isolation_spacing_requirement specifies the Ee_requirement_occurrence (see 4.2.330) of electrical isolation for the Electrical_isolation_removal_structure.

NOTE EXPRESS specification:

```

*)
ENTITY electrical_isolation_removal_structure
  SUBTYPE OF (filled_area_material_removal_structure);
  electrical_isolation_spacing_requirement : ee_requirement_occurrence;
END_ENTITY; -- electrical_isolation_removal_structure

( *

```

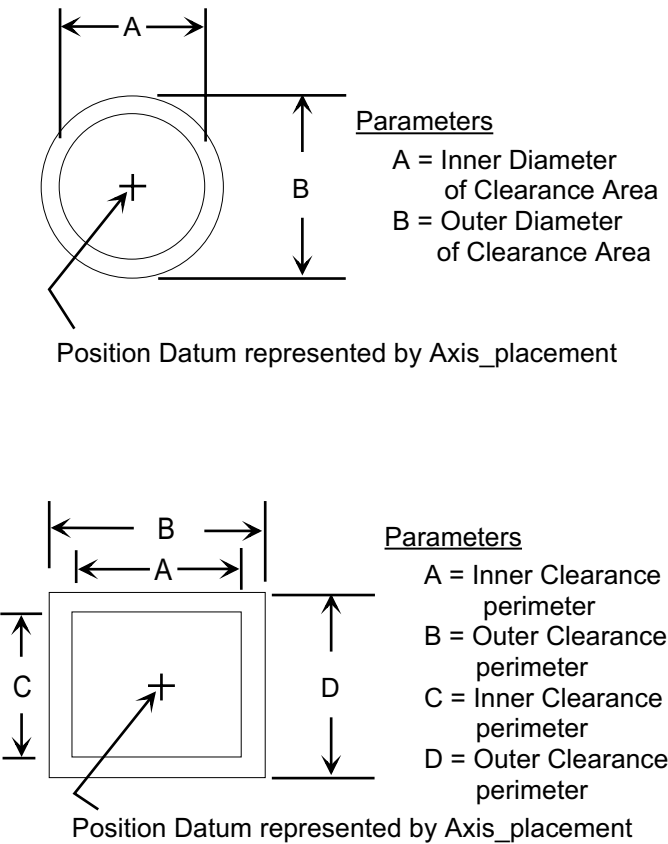
4.2.344 Electrical_isolation_removal_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Electrical_isolation_removal_template is a type of Material_removal_feature_template (see 4.2.534) that describes the shape of material to be removed from a feature. The purposes predefined in this part of ISO 10303 include electrical and thermal isolation.

EXAMPLE 1 Figure 108 illustrates a template and associated parameters designed for electrical isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values.

EXAMPLE 2 Figure 109 illustrates a template and associated parameters designed for thermal isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values. In this example, an instance would be a complex instance including Composite_unit_shape_element (see 4.2.167) and Electrical_isolation_removal_template.

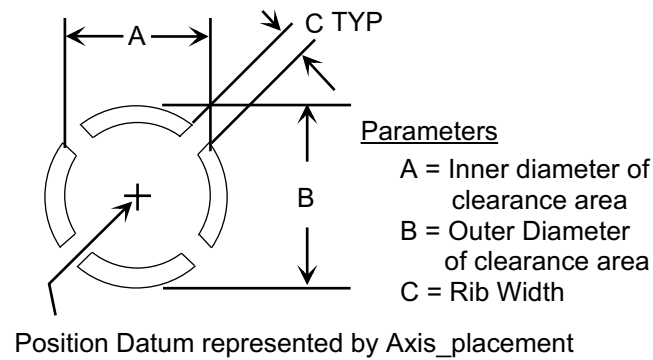
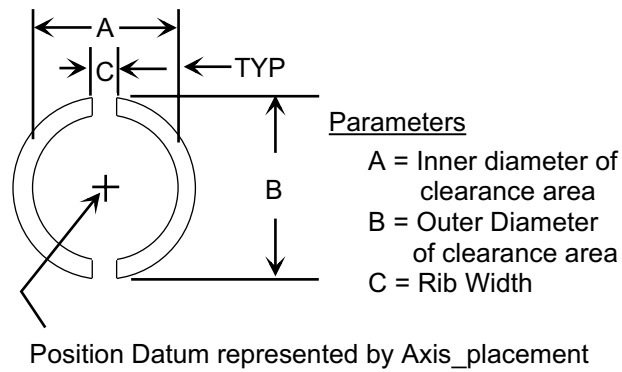


Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Electrical Isolation

Figure 43 – Electrical_isolation_removal_template supports electrical isolation



Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Thermal isolation

Figure 44 – Electrical_isolation_removal_template supports thermal isolation and electrical conduction

The data associated with an `Electrical_isolation_removal_template` are the following:

— `electrical_isolation_spacing_requirement`.

4.2.344.1 `electrical_isolation_spacing_requirement`

The `electrical_isolation_spacing_requirement` specifies the role of the `Layout_spacing_requirement_occurrence` (see 4.2.509) for the `Add_design_object_management_relationship`.

NOTE EXPRESS specification:

```
*)
ENTITY electrical_isolation_removal_template
  SUBTYPE OF (material_removal_feature_template);
  electrical_isolation_spacing_requirement :
    layout_spacing_requirement_occurrence;
END_ENTITY; -- electrical_isolation_removal_template
```

(*

4.2.345 `Electrical_network_definition`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Electrical_network_definition` is a type of `Functional_unit_network_definition` (see 4.2.402) that identifies that the nodes in the network definition default to voltage as the nodal property.

NOTE EXPRESS specification:

```
*)
ENTITY electrical_network_definition
  SUBTYPE OF (functional_unit_network_definition);
END_ENTITY; -- electrical_network_definition
```

(*

An approved `electrical_network` shall be consistent with conservation laws (Kirchoff).

(*

4.2.346 `Electrical_requirement_occurrence`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Electrical_requirement_occurrence` is a type of `Ee_requirement_occurrence` (see 4.2.330) that is related to performance with respect to electrical functionality. An `Electrical_requirement_occurrence` may be a `Material_electrical_conductivity_requirement_occurrence` (see 4.2.532).

NOTE EXPRESS specification:

```
*)
ENTITY electrical_requirement_occurrence
    SUPERTYPE OF (material_electrical_conductivity_requirement_occurrence)
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- electrical_requirement_occurrence

( *
```

4.2.347 Electromagnetic_compatibility_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Electromagnetic_compatibility_requirement_allocation is a type of Assembly_requirement_allocation (see 4.2.53). An Electromagnetic_compatibility_requirement_allocation may be a Shield_allocation (see 4.2.822).

NOTE EXPRESS specification:

```
*)
ENTITY electromagnetic_compatibility_requirement_allocation
    SUPERTYPE OF (shield_allocation)
    SUBTYPE OF (assembly_requirement_allocation);
END_ENTITY; -- electromagnetic_compatibility_requirement_allocation

( *
```

4.2.348 Electromagnetic_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Electromagnetic_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that is related to performance with respect to electromagnetic compatibility and interference.

NOTE EXPRESS specification:

```
*)
ENTITY electromagnetic_requirement_occurrence
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- electromagnetic_requirement_occurrence

--tt tc1 correction
( *
```

4.2.349 Electromotive_force_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Electromotive_force_data_element is a

The data associated with an Electromotive_force_data_element are the following:

— additional_data.

4.2.349.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY electromotive_force_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- electromotive_force_data_element

( *
```

4.2.350 Elliptic_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Elliptic_area is a type of Primitive_2d (see 4.2.720).

The data associated with an Elliptic_area are the following:

— position;

— semi_axis_1;

— semi_axis_2.

4.2.350.1 position

The position specifies an Axis_placement_2d (see 4.2.57) for the Elliptic_area.

4.2.350.2 semi_axis_1

The semi_axis_1 specifies a Length_data_element (see 4.2.511) for the Elliptic_area.

4.2.350.3 semi_axis_2

The semi_axis_2 specifies a Length_data_element (see 4.2.511) for the Elliptic_area.

NOTE EXPRESS specification:

```
*)
ENTITY elliptic_area
  SUBTYPE OF (primitive_2d);
    position : axis_placement_2d;
    semi_axis_1 : length_data_element;
    semi_axis_2 : length_data_element;
END_ENTITY; -- elliptic_area

( *
```

4.2.351 Embedded_component_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Embedded_component_terminal is a type of Laminate_join_terminal (see ??) included in the design to mate directly with a terminal of a fabricated feature by use of a Fabrication_joint (see 4.2.368) internally in a layout. An Embedded_component_terminal is a model specifically included to bridge between the discrete purchased items used in a design and the custom fabricated interconnect patterns in a substrate design. An Embedded_component_terminal is an instance of a usage definition for access to a discrete component in the context of a substrate design. An Embedded_component_terminal is included to support using Physical_network to define the implementation connections to the associated component.

NOTE The modeling approach used here does not imply that Embedded_component_terminal is created as part of the wet or dry chemistr process used to create the patterns on the substrate. It is merely identifying a functional role of a terminal of a discrete part when it is embedded in a substrate. By implication, it is also identifying the role of the associated component, i.e. that the associated component is embedded in the substrate. The method or process of achieving this is not defined in this standard. The Embedded_component_terminal will not be included in a Physical_connectivity_definition and is treated herein similarly to the Printed_component terminals. A Fabrication_joint is required to connect the terminal to the Land_join_terminal adjacent to satisfy functional connectivity requirements.

The data associated with an Embedded_component_terminal are the following:

- associated_component;
- definition.

4.2.351.1 associated_component

The associated_component specifies the role of the Bare_die_component (see 4.2.61) or the role of the Packaged_component (see 4.2.585) for the Embedded_component_terminal.

4.2.351.2 definition

The definition specifies the role of the Bare_die_terminal (see 4.2.66) or the role of the Packaged_part_terminal (see 4.2.594) for the Embedded_component_terminal.

Constraints:

WR1: An Embedded_component_terminal shall participate in a relationship with a Fabrication_joint as a fabrication_features.

NOTE EXPRESS specification:

```
*)
ENTITY embedded_component_terminal
  SUBTYPE OF (laminate_component_join_terminal);
  SELF\component_feature.definition      : bare_die_terminal_or_packaged_part_termi
  SELF\component_feature.associated_component : bare_die_component_or_packaged_component
END_ENTITY; -- embedded_component_terminal

( *
```

4.2.352 Energy_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Energy_data_element is a

The data associated with an Energy_data_element are the following:

— additional_data.

4.2.352.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
* )
ENTITY energy_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- energy_data_element

( *
```

4.2.353 Engineering_make_from

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Engineering_make_from is a type of Managed_design_object (see 4.2.524). An Engineering_make_from is the relationship between two products in which one product is used as the basis for the design of the other product.

EXAMPLE 1 Engineering_make_from may be mask programmed processors, enterprise programmed digital devices, or field programmable or field reprogrammable arrays or modems that are updateable by downloading new firmware from a central source.

NOTE This facility is not limited to packaged parts but may be applied to any product.

EXAMPLE 2 A mask programmed processor would be a 'mask programmable processor' before programming and would be a 'Yaw rate processor' after programming.

The data associated with an Engineering_make_from are the following:

- base_design;
- resultant_design.

4.2.353.1 base_design

The base_design specifies the Ee_product_definition (see 4.2.323) to be used as the basis for the design of another Ee_product (see 4.2.319) for the Engineering_make_from.

4.2.353.2 resultant_design

The resultant_design specifies the Ee_product_definition (see 4.2.323) that results from basing the design upon the design of another Ee_product (see 4.2.319) for the Engineering_make_from.

NOTE EXPRESS specification:

```

*)
ENTITY engineering_make_from
  SUBTYPE OF (managed_design_object);
    base_design      : ee_product_definition;
    resultant_design : ee_product_definition;
END_ENTITY; -- engineering_make_from

( *

```

4.2.354 Equivalent_functional_terminals_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Equivalent_functional_terminals_assignment is a type of Managed_design_object (see 4.2.524). An Equivalent_functional_terminals_assignment is the assignment that a set of terminals are equivalent in a Functional_unit_usage_view (see 4.2.410).

The data associated with an Equivalent_functional_terminals_assignment are the following:

- equivalent_terminals;
- name.

4.2.354.1 equivalent_terminals

The equivalent_terminals specify the equivalent Functional_unit_usage_view_terminal_definition (see 4.2.411) for the Equivalent_functional_terminals_assignment. There shall be at least two Functional_unit_usage_view_terminal_definition for the Equivalent_functional_terminals_assignment.

4.2.354.2 name

The name specifies the string that is the identifier for the Equivalent_functional_terminals_assignment.

Constraints:

WR1: One member of equivalent_functional_terminals_assignment shall establish equivalency only between members of functional_unit_usage_view_terminal_definition that are terminals of the same functional_unit_definition.

NOTE EXPRESS specification:

```

*)
ENTITY equivalent_functional_terminals_assignment
  SUBTYPE OF (managed_design_object);
    equivalent_terminals : SET [2:?] OF functional_unit_usage_view_terminal_definition;
    name                  : ee_name;
END_ENTITY; -- equivalent_functional_terminals_assignment

( *

```

4.2.355 Equivalent_functional_unit_definition_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Equivalent_functional_unit_definition_assignment is a type of Managed_design_object (see 4.2.524). An Equivalent_functional_unit_definition_assignment is the association between at least two Functional_unit_definition (see 4.2.401) that are equivalent.

The data associated with an Equivalent_functional_unit_definition_assignment are the following:

- equivalent_functional_unit_definitions;
- name.

4.2.355.1 equivalent_functional_unit_definitions

The equivalent_functional_unit_definitions specify the Functional_unit_definition (see 4.2.401) for the Equivalent_functional_unit_definition_assignment. There shall be at least two but may be more than two Functional_unit_definition for the Equivalent_functional_unit_definition_assignment.

4.2.355.2 name

The name specifies the string that is the identifier for the Equivalent_functional_unit_definition_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY equivalent_functional_unit_definition_assignment
  SUBTYPE OF (managed_design_object);
    equivalent_functional_unit_definitions : SET [2:?] OF functional_unit_definition;
    name                                  : ee_name;
END_ENTITY; -- equivalent_functional_unit_definition_assignment

( *

```

4.2.356 Evaluated_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Evaluated_characteristic is a type of Managed_design_object (see 4.2.524). The Evaluated_characteristic provides a mechanism to validate or evaluate a Planned_characteristic (see 4.2.679).

The data associated with an Evaluated_characteristic are the following:

- planned_product_data;
- product_characterization_data;
- product_data_status.

4.2.356.1 planned_product_data

The planned_product_data specifies the role of the Planned_characteristic (see 4.2.679) for an Evaluated_characteristic.

4.2.356.2 product_characterization_data

The product_characterization_data specifies one or more Coordinated_characteristic (see 4.2.197) for an Evaluated_characteristic. The product_characterization_data need not be provided for a specific Evaluated_characteristic.

4.2.356.3 product_data_status

The product_data_status provides a status of valid, invalid, not_evaluable, or to_be_evaluated for an Evaluated_characteristic.

NOTE EXPRESS specification:

```

*)
ENTITY evaluated_characteristic
  SUPERTYPE OF (rule_based_evaluated_characteristic)
  SUBTYPE OF (managed_design_object);
    planned_product_data          : planned_characteristic;
    product_data_status           : evaluation_type;
    product_characterization_data : OPTIONAL SET [1:?] OF coordinated_characteristic;
    evaluation_result              : evaluation_result_value;
END_ENTITY; -- evaluated_characteristic
( *
```

4.2.357 Evaluation_ee_product_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Evaluation_ee_product_definition is a type of Ee_product_definition (see 4.2.323) that contains the results of evaluating another Ee_product_definition.

The data associated with an Evaluation_ee_product_definition are the following:

- evaluated_product_definition;
- evaluation_data.

4.2.357.1 evaluated_product_definition

The evaluated_product_definition specifies the Ee_product_definition that was evaluated.

4.2.357.2 evaluation_data

The evaluation_data specifies the role of the Evaluated_characteristic (see 4.2.356) that contains the detailed product data resulting from the evaluation.

NOTE EXPRESS specification:

```
*)
ENTITY evaluation_ee_product_definition
  SUBTYPE OF (ee_product_definition);
    evaluated_product_definition : ee_product_definition;
    evaluation_data : SET [1:?] OF evaluated_characteristic;
WHERE
  WR1: TYPEOF(SELF) &#60;&#62; TYPEOF(evaluated_product_definition);
END_ENTITY; -- evaluation_ee_product_definition
(*
```

4.2.358 Event_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Event_data_element is a

The data associated with an Event_data_element are the following:

- additional_data.

4.2.358.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY event_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- event_data_element

( *
```

4.2.359 Expanded_uncertainty

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Expanded_uncertainty is a type of Standard_uncertainty (see 4.2.837). The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY expanded_uncertainty
  SUBTYPE OF (standard_uncertainty);
  coverage_factor : REAL;
END_ENTITY;
( *
```

4.2.360 Explicitly_located_layer_connection_point

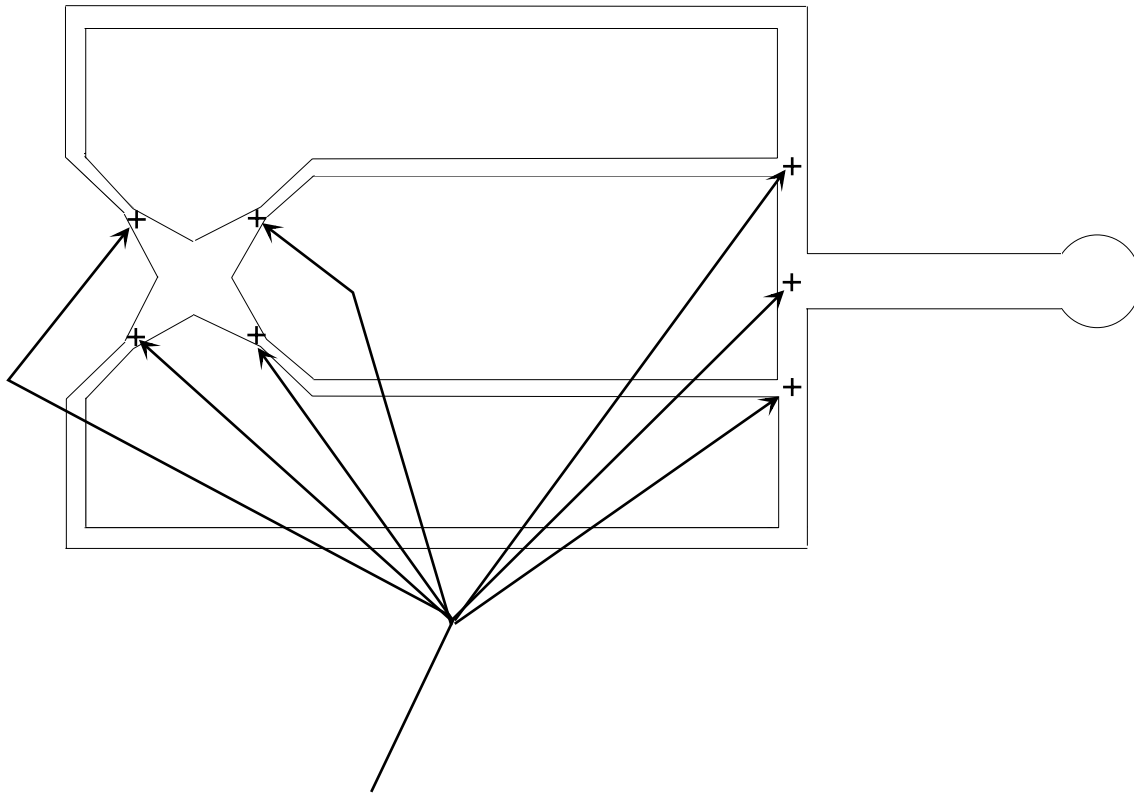
NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Explicitly_located_layer_connection_point is a type of Layer_connection_point (see 4.2.499) that is not a Dependently_located_layer_connection_point (see 4.2.259). An Explicitly_located_layer_connection_point may be referenced by a Junction_vertex_allocation to help define a topological structure.

EXAMPLE 1 Figure 45 illustrates a detail of a layout drawing that shows the Explicitly_located_layer_connection_points that are included in the design due to internal layout considerations.

The data associated with an Explicitly_located_layer_connection_point are the following:

— location.



"+" Explicitly_located_layer_connection_point

Figure 45 – Explicitly_located_layer_connection_point

4.2.360.1 location

The location specifies the Cartesian_point (see 4.2.87) relative to the Cartesian_point origin of the Interconnect_module (see 4.2.440) at which the Explicitly_located_layer_connection_point resides.

Constraints:

UR1: The combination of location, and resident_design_layer_stratum inherited from the supertype Layer_connection_point shall be unique within a population of Explicitly_located_layer_connection_point.

WR1: The location shall be in the extent of the resident_design_layer_stratum.

NOTE EXPRESS specification:

```

*)
ENTITY explicitly_located_layer_connection_point
  SUBTYPE OF (layer_connection_point);
(*
--tt tc1 wd9 moved to supertype
  location : cartesian_point;
  UNIQUE
    UR1 : location, SELF\layer_connection_point.resident_design_layer_stratum;
*)
END_ENTITY; -- explicitly_located_layer_connection_point

(*

```

4.2.361 External_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An External_definition is a type of Managed_design_object (see 4.2.524). An External_definition is a definition that is supported externally to this part of ISO 10303.

The data associated with an External_definition are the following:

- code;
- code_value;
- reference_document.

4.2.361.1 code

The code specifies the string as the identifier of the type of the External_definition.

4.2.361.2 code_value

The code_value specifies the text as the content of the External_definition.

4.2.361.3 reference_document

The reference_document specifies either the Ee_document (see 4.2.316) or a string referred to as a standard document identification string, registered by IEC or ISO for the purpose of unambiguously identifying a document published by the respective organization, as the code definition document used by application programmers for the External_definition.

NOTE The reference_document includes a data dictionary that defines the allowed types that may be used as identifiers.

Constraints:

IP1: The string specified by code shall not appear more than once in the reference_document. This constraint is to be interpreted as a requirement against the implementation of document when that implementation is to support External_definition.

NOTE EXPRESS specification:

```
*)
ENTITY external_definition
  SUBTYPE OF (managed_design_object);
  code          : STRING;
  code_value    : STRING;
  reference_document : standard_or_reference_document;
END_ENTITY; -- external_definition
```

(*

4.2.362 Externally_defined_curve_font

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Externally_defined_curve_font is a type of Managed_design_object (see 4.2.524). An Externally_defined_curve_font is a curve font that is defined in an external computer interpretable system.

NOTE EXPRESS specification:

```
*)
ENTITY externally_defined_curve_font
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_curve_font
```

(*

4.2.363 Externally_defined_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Externally_defined_dimension_representation is a type of Shape_dimension_representation (see 4.2.809) the interpretation of which is dependent on definitions supported externally to this part of ISO 10303.

The data associated with an Externally_defined_dimension_representation are the following:

- definitions;
- dimension_value.

4.2.363.1 definitions

The definitions specify the External_definition (see 4.2.361) that represents the Dimensional_characteristic (see 4.2.298). There shall be one or more External_definition specified as definitions for an Externally_defined_dimension_representation.

4.2.363.2 dimension_value

The dimension_value specifies the Data_element (see 4.2.211) that in conjunction with the External_definition (see 4.2.361), that are specified as the definitions, represent the associated Dimensional_characteristic (see 4.2.298). The dimension_value need not be specified for an Externally_defined_dimension_representation.

EXAMPLE 1 The system of limits and fits specifies a way in which to represent dimensions and is defined in ISO 286.

NOTE EXPRESS specification:

```
*)
ENTITY externally_defined_dimension_representation
  SUBTYPE OF (shape_dimension_representation);
  definitions      : SET [1:?] OF external_definition;
  dimension_value : OPTIONAL data_element;
END_ENTITY; -- externally_defined_dimension_representation

( *
```

4.2.364 Externally_defined_ee_product_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Externally_defined_ee_product_definition is a type of Ee_product_definition (see 4.2.323) that is defined in an external computer interpretable system.

The data associated with an Externally_defined_ee_product_definition are the following:

— source.

4.2.364.1 source

The source specifies the name of the source of the Externally_defined_ee_product_definition.

NOTE EXPRESS specification:

```
*)  
ENTITY externally_defined_ee_product_definition  
  SUBTYPE OF (ee_product_definition);  
  source : ee_name;  
END_ENTITY; -- externally_defined_ee_product_definition  
  
(*
```

4.2.365 Externally_defined_hatch_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Externally_defined_hatch_style is a type of Managed_design_object (see 4.2.524). An Externally_defined_hatch_style is a hatch style that is defined in an external computer interpretable system.

NOTE EXPRESS specification:

```
*)  
ENTITY externally_defined_hatch_style  
  SUBTYPE OF (managed_design_object);  
END_ENTITY; -- externally_defined_hatch_style  
  
(*
```

4.2.366 Externally_defined_tile

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Externally_defined_tile is a type of Managed_design_object (see 4.2.524). An Externally_defined_tile is a tile that is defined in an external computer interpretable system.

NOTE EXPRESS specification:

```
*)  
ENTITY externally_defined_tile  
  SUBTYPE OF (managed_design_object);  
END_ENTITY; -- externally_defined_tile  
  
(*
```

4.2.367 Externally_defined_tile_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Externally_defined_tile_style` is a type of `Managed_design_object` (see 4.2.524). An `Externally_defined_tile_style` is a tile style that is defined in an external computer interpretable system.

NOTE EXPRESS specification:

```
*)
ENTITY externally_defined_tile_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_tile_style

( *
```

4.2.368 Fabrication_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Fabrication_joint` is a type of `Managed_design_object` (see 4.2.524). A `Fabrication_joint` is a joint that is created in the fabrication of a `Interconnect_module` (see 4.2.440).

EXAMPLE 1 Figure 46 is a plan view of the construction of a `Fabrication_joint` between a `Land` (see 4.2.490) and a `Conductive_interconnect_element` (see 4.2.171) that results in a `Conductor` (see 4.2.176) segment. This figure illustrates the geometric relationship of the element terminals in the construction of the segment.

EXAMPLE 2 Figure 47 is a cross-sectional view of the construction of a `Fabrication_joint` between a `Land` and a `Component_termination_passage` (see 4.2.143).

The data associated with a `Fabrication_joint` are the following:

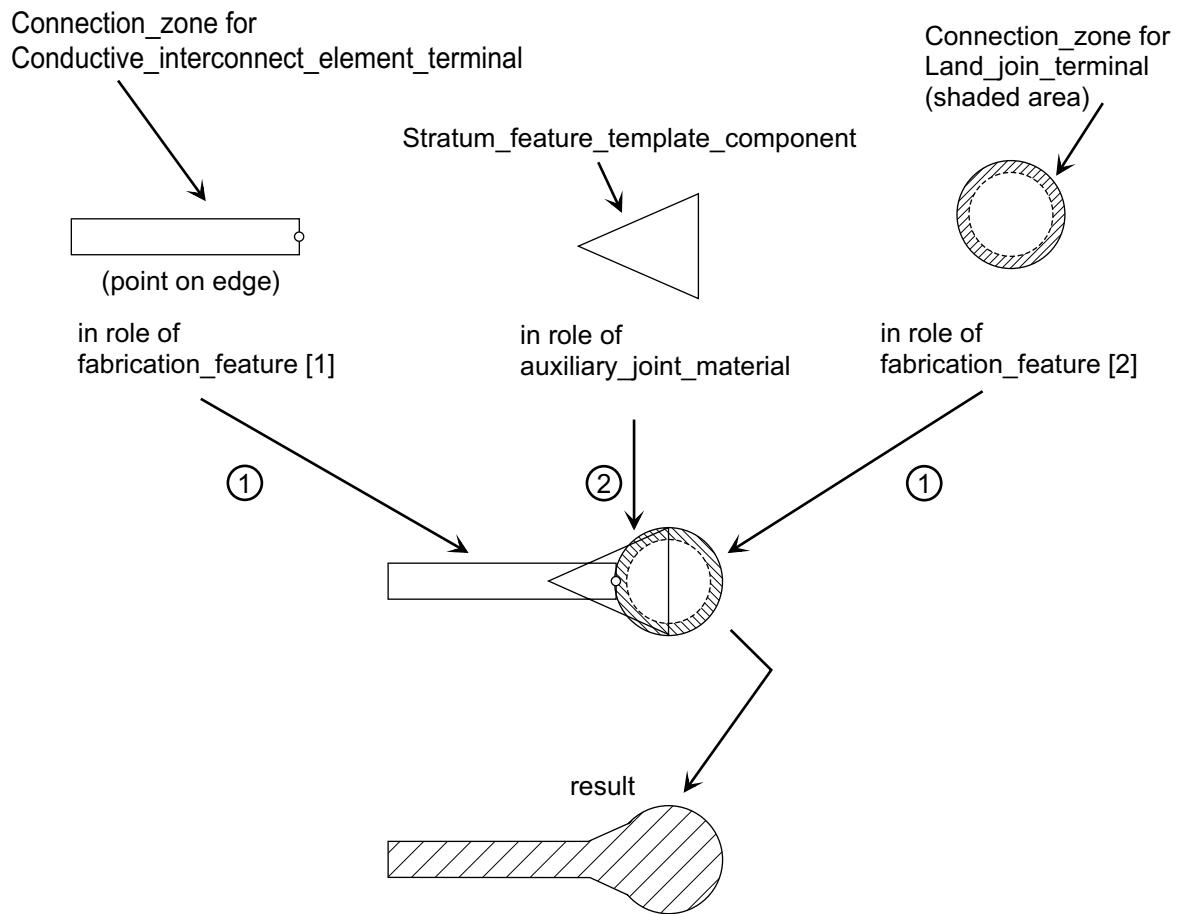
- `auxiliary_joint_material`;
- `fabrication_features`.

4.2.368.1 auxiliary_joint_material

The `auxiliary_joint_material` specifies the `Stratum_feature_template_component` (see 4.2.862) for the `Fabrication_joint`. The `auxiliary_joint_material` need not be specified for a particular `Fabrication_joint`.

4.2.368.2 fabrication_features

The `fabrication_features` specify exactly two from among the following: a `Component_termination_passage_join_terminal`, a `Conductive_interconnect_element_terminal` (see 4.2.172), a `Land_join_terminal` (see 4.2.492), a `Non_functional_land_join_terminal` (see 4.2.564), a `Printed_component_join_terminal` (see 4.2.722), or a `Via_terminal` (see 4.2.954) for the `Fabrication_joint`.



Conductor segment resulting from Fabrication_joint.

* Step 1 : relate 2 terminals (of already placed components)

* Step 2 : relate a Stratum_feature_template_component to the joint

Note : 1) This supports filleting and keyholing at the intersection of Land and trace
Reference IPC-D-275.

2) The design view is the Interconnect_module.

Figure 46 – Land to Conductive_interconnect_element Fabrication_joint exploded view

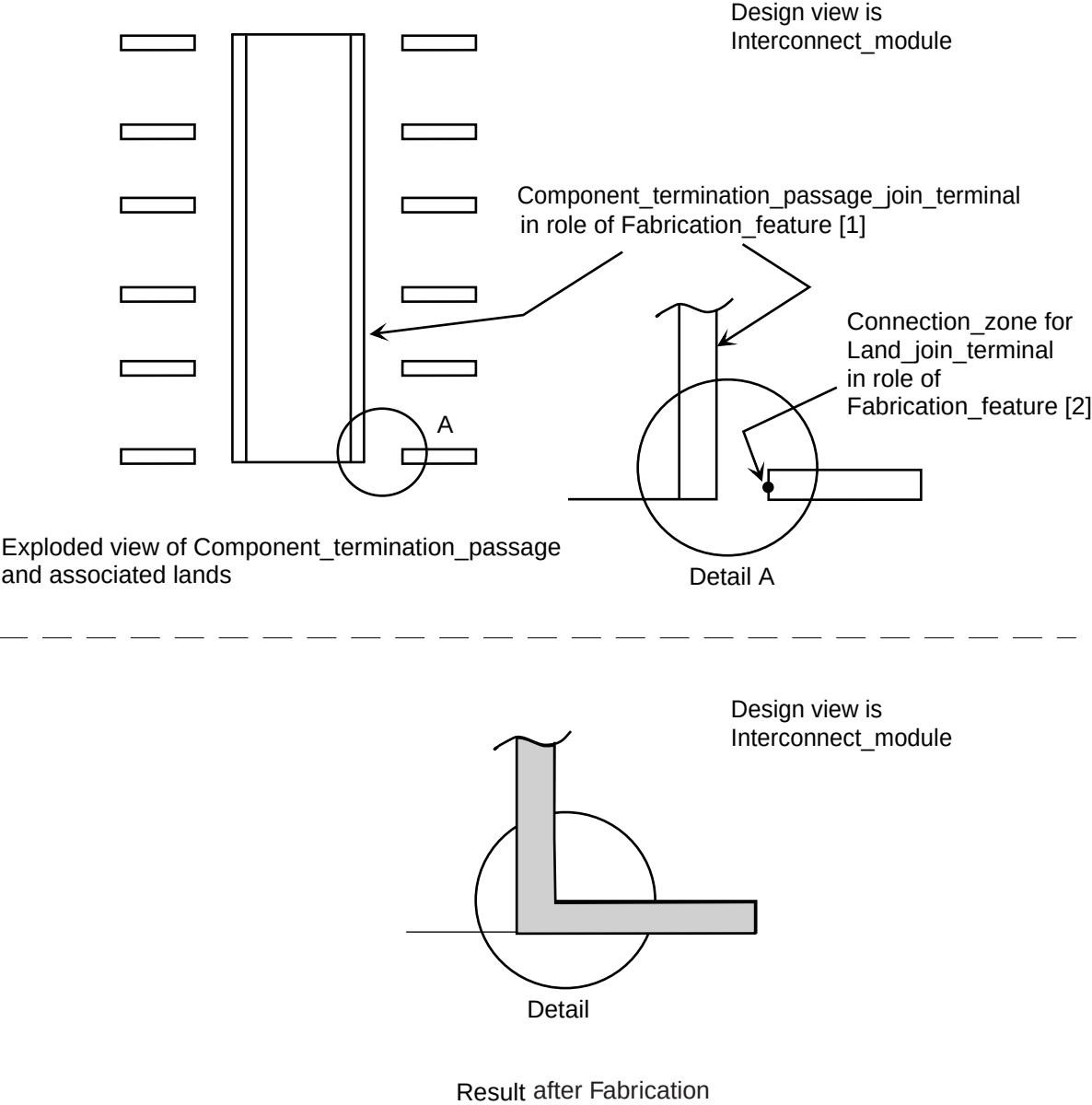


Figure 47 – Land to Component_termination_passage exploded view

Constraints:

WR1: A member of `Conductive_interconnect_element_terminal` and a member of `Non_functional_land_join_terminal` shall not be in the same set of `fabrication_features`.

NOTE EXPRESS specification:

```
*)
ENTITY fabrication_joint
  SUBTYPE OF (managed_design_object);
    fabrication_features_1 : fabricated_feature_select;
    fabrication_features_2 : fabricated_feature_select;
    auxiliary_joint_material : OPTIONAL stratum_feature_template_component;
    associated_interconnect_module : interconnect_module;
END_ENTITY; -- fabrication_joint
```

(*

4.2.369 Fabrication_technology_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Fabrication_technology_specification` is a type of `Process_specification` (see 4.2.740) that describes the technology used for parts and part families.

NOTE EXPRESS specification:

```
*)
ENTITY fabrication_technology_specification
  SUBTYPE OF (process_specification);
END_ENTITY; -- fabrication_technology_specification
```

(*

4.2.370 Fabrication_thickness_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Fabrication_thickness_requirement_occurrence` is a type of `Inspection_requirement_occurrence` (see 4.2.430) that characterizes the interconnect thickness range over metallic material or dielectric material and assigns a tolerance to that range for fabrication purposes.

The data associated with a `Fabrication_thickness_requirement_occurrence` are the following:

- `characterized_class`;
- `characterized_class_range`;
- `tolerance`.

4.2.370.1 `characterized_class`

The `characterized_class` specifies either "over_finished_conductor" or "over_finished_dielectric" for the `Fabrication_thickness_requirement_occurrence`. This is the class to which the range and tolerance is applied.

4.2.370.2 `characterized_class_range`

The `characterized_class_range` specifies the role of the `Range_characteristic` (see 4.2.760) for the `Fabrication_thickness_requirement_occurrence`. The `characterized_class_range` specifies the set of measurements to which the tolerance applies.

4.2.370.3 `tolerance`

The `tolerance` specifies the role of the `Tolerance_characteristic` (see 4.2.911) for the `Fabrication_thickness_requirement_occurrence`. The tolerance is applicable over the range specified by the `characterized_class_range`.

NOTE EXPRESS specification:

```
*)
ENTITY fabrication_thickness_requirement_occurrence
  SUBTYPE OF (inspection_requirement_occurrence);
    characterized_class      : metal_or_dielectric;
    characterized_class_range : range_characteristic;
    tolerance                 : tolerance_characteristic;
END_ENTITY; -- fabrication_thickness_requirement_occurrence
(*
```

4.2.371 `Feature_group_relationship`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Feature_group_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Feature_group_relationship` is the association of one or more `Part_feature` (see 4.2.609) and a `Part_group_feature` (see 4.2.613) that are used in the composition of that `Part_group_feature`.

The data associated with a Feature_group_relationship are the following:

- related_feature;
- relating_feature.

4.2.371.1 related_feature

The related_feature specifies the Part_feature (see 4.2.609) used in the Feature_group_relationship.

4.2.371.2 relating_feature

The relating_feature specifies the Part_group_feature (see 4.2.613) in the Feature_group_relationship.

NOTE EXPRESS specification:

```
*)  
ENTITY feature_group_relationship  
  SUBTYPE OF (managed_design_object);  
    relating_feature : part_group_feature;  
    related_feature  : part_feature;  
END_ENTITY; -- feature_group_relationship
```

(*

4.2.372 Feature_shape_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Feature_shape_definition is a type of Shape_definition (see 4.2.808). A Feature_shape_definition represents real features in the product model. A member of Feature_shape_definition shall be also be a member of another subtype of Shape_definition.

NOTE The type constraint stated is supported by the information requirements described in the Shape_definition Application Object.

NOTE EXPRESS specification:

```
*)  
ENTITY feature_shape_definition  
  SUBTYPE OF (shape_definition);  
END_ENTITY; -- feature_shape_definition  
( *
```

4.2.373 Fiducial

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fiducial is a type of Laminate_component (see 4.2.483) that is a basis of measurement with respect to other Assembly_components. A Fiducial is used for accurate alignment of one item with another. Another type of Laminate_component may play the role of Fiducial in a design.

NOTE EXPRESS specification:

```
*)
ENTITY fiducial
  SUBTYPE OF (laminate_component);
END_ENTITY; -- fiducial
```

(*

4.2.374 Fiducial_part_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fiducial_part_feature is a type of Part_tooling_feature (see 4.2.633). A Fiducial_part_feature identifies a feature for product alignment with respect to a tool.

NOTE EXPRESS specification:

```
*)
ENTITY fiducial_part_feature
  SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- fiducial_part_feature
```

(*

4.2.375 Fiducial_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fiducial_stratum_feature is a type of Stratum_feature (see 4.2.851) that is a basis of measurement with respect to other Stratum_features. A Fiducial_stratum_feature is used for accurate alignment of one item with another. Another type of Stratum_feature may play the role of Fiducial_stratum_feature in a design.

NOTE EXPRESS specification:

```
*)
ENTITY fiducial_stratum_feature
  SUBTYPE OF (stratum_feature);
END_ENTITY; -- fiducial_stratum_feature

--tt tc1 wd9 modified.
(*
```

4.2.376 Fill_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area is a type of Managed_design_object (see 4.2.524). A Fill_area is the area bounded by a Curve (see 4.2.203) that is defined to be filled with material using a pattern. Each Curve shall be an outer or inner boundary. Each boundary shall be closed. There shall be no more than one outer boundary. No boundary shall self-intersect or intersect with any other boundary.

NOTE This Application Object is describing geometric patterns. The association of material with the patterns occurs elsewhere, for instance in the Design_intent_modification_styled_area_representation (see 4.2.274).

The data associated with a Fill_area are the following:

— boundary.

4.2.376.1 boundary

The boundary specifies the Closed_curve (see 4.2.104) that describes the outline of the Fill_area. There shall be one or more Curve for a Fill_area. The curves shall be non-self intersecting and non-overlapping.

NOTE EXPRESS specification:

```
*)
ENTITY fill_area
  SUBTYPE OF (managed_design_object);
  boundaries : SET [1:?] OF closed_curve;
END_ENTITY; -- fill_area

(*
```

4.2.377 Fill_area_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_occurrence is a type of Managed_design_object (see 4.2.524). A Fill_area_occurrence is an instantiation of a Fill_area (see 4.2.376). The boundary_style shall be applied to each boundary of the instance of the Fill_area. A zero width boundary_style shall be interpreted as using the boundaries of the Fill_area as clipping regions. A non-zero width boundary_style shall be interpreted as the regularized Boolean "OR" operation on the instance of the Fill_area, applied at each boundary after the clip operation required by the zero width boundary style. The orientation of the Fill_area instances shall be parallel with the orientation of the Fill_area.

The data associated with a Fill_area_occurrence are the following:

- boundary_style;
- filled_area;
- fill_style_target;
- style.

4.2.377.1 boundary_style

The boundary_style specifies the role of the Curve_style (see 4.2.205) for the Fill_area_occurrence. The boundary_style is applied to each instance of the Fill_area boundaries individually. The boundary_style end parameters shall be ignored. The boundary_style curve_font shall not be specified.

4.2.377.2 filled_area

The filled_area specifies the role of the Fill_area (see 4.2.376) for the Fill_area_occurrence. The Fill_area provides shape and orientation information for the Fill_area_occurrence.

4.2.377.3 fill_style_target

The fill_style_target specifies the Cartesian_point (see 4.2.87) that defines the location of the Fill_area_style (see 4.2.378) for the Fill_area_occurrence.

4.2.377.4 style

The style specifies the role of the Fill_area_style (see 4.2.378) for the Fill_area_occurrence. There shall be one or more Fill_area_style for a Fill_area_occurrence.

NOTE EXPRESS specification:

```
*)  
ENTITY fill_area_occurrence  
  SUBTYPE OF (managed_design_object);  
    style          : SET [1:?] OF fill_area_style;  
    filled_area     : fill_area;  
    fill_style_target : cartesian_point;  
    boundary_style  : curve_style;  
END_ENTITY; -- fill_area_occurrence  
  
(*
```

4.2.378 Fill_area_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_style is a type of Managed_design_object (see 4.2.524). A Fill_area_style is the selected style used in a Fill_area (see 4.2.376).

The data associated with a Fill_area_style are the following:

- fill_style;
- name.

4.2.378.1 fill_style

The fill_style specifies either an Externally_defined_hatch_style (see 4.2.365), an Externally_defined_tile_style (see 4.2.367), a Fill_area_style_hatching (see 4.2.379), a Fill_area_style_tile (see 4.2.380), or a Solid_fill_style (see 4.2.832) for the Fill_area_style.

4.2.378.2 name

The name specifies the string for the identifier for the Fill_area_style.

Constraints:

UR1: The name shall be unique within a population of Fill_area_style.

NOTE EXPRESS specification:

```

*)
ENTITY fill_area_style
  SUBTYPE OF (managed_design_object);
    name      : STRING;
    fill_style : fill_style_select;
  UNIQUE
    UR1: name;
END_ENTITY; -- fill_area_style

( *
```

4.2.379 Fill_area_style_hatching

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_style_hatching is a type of Managed_design_object (see 4.2.524). A Fill_area_style_hatching is the parallel line hatch style of a Fill_area (see 4.2.376). All units associated with Fill_area_style_hatching are defined in the global units for the Interconnect_module (see 4.2.440) being designed. The geometric reference for Fill_area_style_hatching shall be the origin of the Interconnect_module being designed. When this Application Object is used to help compose a Design_intent_modification_styled_area_representation (see 4.2.274) or to help compose a Stratum_feature_styled_area_representation (see 4.2.860), only the Curve_style_font (see 4.2.206) shall be referenced by the curve_font attribute of the Curve_style Application Object referenced by hatch_line_appearance.

NOTE In order to obtain cross-hatch, two instances of Fill_area_style_hatching are required, with complementary angles.

The data associated with a Fill_area_style_hatching are the following:

- hatch_line_angle;
- hatch_line_appearance;
- pattern_start;
- point_of_reference_hatch_line;
- start_of_next_hatch_line.

4.2.379.1 hatch_line_angle

The hatch_line_angle specifies the role of the Angle_data_element (see 4.2.23) for the Fill_area_style_hatching.

4.2.379.2 hatch_line_appearance

The hatch_line_appearance specifies the role of the Curve_style (see 4.2.205) for the Fill_area_style_hatching.

NOTE The curve style may include a discontinuous geometric shape, in which case the hatch cannot be used to represent product data which is contiguous in nature.

EXAMPLE 1 The curve style may be a dashed line and in that case each of the dashes would correspond to a different Stratum_feature (see 4.2.851).

4.2.379.3 pattern_start

The pattern_start specifies the Cartesian_point (see 4.2.87) for the beginning of the pattern of Fill_area_style_hatching.

4.2.379.4 point_of_reference_hatch_line

The point_of_reference_hatch_line specifies the Cartesian_point (see 4.2.87) for the point of reference and located on the initial hatch line for the Fill_area_style_hatching.

4.2.379.5 start_of_next_hatch_line

The start_of_next_hatch_line specifies the Vector (see 4.2.946) from the end Cartesian_point of the present hatch line to the beginning Cartesian_point of the next hatch line for the Fill_area_style_hatching.

NOTE EXPRESS specification:

```
*)
ENTITY fill_area_style_hatching
  SUBTYPE OF (managed_design_object);
    hatch_line_appearance      : curve_style;
    start_of_next_hatch_line   : vector;
    point_of_reference_hatch_line : cartesian_point;
    pattern_start              : cartesian_point;
    hatch_line_angle           : angle_data_element;
END_ENTITY; -- fill_area_style_hatching
```

(*

4.2.380 Fill_area_style_tile

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_style_tile is a type of Managed_design_object (see 4.2.524). A Fill_area_style_tile is the definition of the tile for a Fill_area_style (see 4.2.378).

The data associated with a `Fill_area_style_tile` are the following:

- `tile`;
- `tiling_pattern`;
- `tiling_scale`.

4.2.380.1 `tile`

The `tile` specifies either an `Externally_defined_tile` (see 4.2.366), a `Fill_area_style_tile_curve_with_style` (see 4.2.381), a `Fill_area_style_tile_solid_region` (see 4.2.382), or a `Fill_area_style_tile_symbol_with_style` (see 4.2.383) for the `Fill_area_style_tile`.

4.2.380.2 `tiling_pattern`

The `tiling_pattern` specifies two `Vector` (see 4.2.946) for the `Fill_area_style_tile`. There shall be exactly two `Vector` for a `Fill_area_style_tile`.

4.2.380.3 `tiling_scale`

The `tiling_scale` specifies the role of the `Positive_ratio_measure` (see ??) for a `Fill_area_style_tile`.

NOTE EXPRESS specification:

```

*)
ENTITY fill_area_style_tile
  SUBTYPE OF (managed_design_object);
  --tt tc1 wd11 modified
    first_direction_repeat_factor : vector;
  --tt tc1 wd11 added
    second_direction_repeat_factor : vector;
    tile : fill_area_style_tile_shape_select;
    tiling_scale : positive_ratio_measure;
END_ENTITY; -- fill_area_style_tile

( *
```

4.2.381 `Fill_area_style_tile_curve_with_style`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Fill_area_style_tile_curve_with_style` is a type of `Managed_design_object` (see 4.2.524). A `Fill_area_style_tile_curve_with_style` is the definition of the tiling of a `Fill_area` (see 4.2.376).

NOTE EXPRESS specification:

```
*)  
ENTITY fill_area_style_tile_curve_with_style  
  SUBTYPE OF (managed_design_object);  
END_ENTITY; -- fill_area_style_tile_curve_with_style  
  
( *
```

4.2.382 Fill_area_style_tile_solid_region

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_style_tile_solid_region is a type of Managed_design_object (see 4.2.524). A Fill_area_style_tile_solid_region is the definition of a region that shall be filled solidly with tiles.

NOTE EXPRESS specification:

```
*)  
ENTITY fill_area_style_tile_solid_region  
  SUBTYPE OF (managed_design_object);  
END_ENTITY; -- fill_area_style_tile_solid_region  
  
( *
```

4.2.383 Fill_area_style_tile_symbol_with_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Fill_area_style_tile_symbol_with_style is a type of Managed_design_object (see 4.2.524). A Fill_area_style_tile_symbol_with_style is the definition of a tile symbol with a specified style.

NOTE EXPRESS specification:

```
*)  
ENTITY fill_area_style_tile_symbol_with_style  
  SUBTYPE OF (managed_design_object);  
END_ENTITY; -- fill_area_style_tile_symbol_with_style  
  
( *
```

4.2.384 Filled_area_material_removal_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Filled_area_material_removal_laminate_component` is a type of `Material_removal_laminate_component` (see 4.2.535) that indicates that material is to be removed in the realization of the feature that is in the role of `design_intent` inherited from `Material_removal_laminate_component`. A `Filled_area_material_removal_laminate_component` is either an `Electrical_isolation_laminate_component` (see 4.2.342) or a `Thermal_isolation_laminate_component` (see 4.2.902). The requirement shall be interpreted mathematically as a requirement for application of the regularized Boolean subtraction operation. The `design_intent` inherited from `Material_removal_laminate_component` shall not be a `Conductor` (see 4.2.176).

NOTE In some cases, the isolation may be accomplished by complete removal, as in a dielectric removal from a dielectric layer or layers.

The data associated with a `Filled_area_material_removal_laminate_component` are the following:

— `description`.

4.2.384.1 description

The `reference_padstack` specifies the role of the `Padstack_definition` (see 4.2.595) for the `Filled_area_material_removal_laminate_component`.

NOTE EXPRESS specification:

```

*)
ENTITY filled_area_material_removal_laminate_component
  SUPERTYPE OF (ONEOF(electrical_isolation_laminate_component,
    thermal_isolation_laminate_component))
  SUBTYPE OF (material_removal_laminate_component);
  reference_padstack : padstack_definition;
WHERE
  WR1: NOT('AP210_ARM.CONDUCTOR' IN TYPEOF(SELF\material_removal_laminate_component.design_intent));
END_ENTITY; -- filled_area_material_removal_laminate_component

( *
```

4.2.385 Filled_area_material_removal_structure

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Filled_area_material_removal_structure` is a type of `Material_removal_feature` (see 4.2.533). A `Filled_area_material_removal_structure` is either an `Electrical_isolation_removal_structure` (see 4.2.343) or a `Thermal_isolation_removal_structure` (see 4.2.903).

The data associated with a `Filled_area_material_removal_structure` are the following:

— `design_intent`.

4.2.385.1 design_intent

The design_intent specifies the Conductive_filled_area (see 4.2.170) for the requirement for the Filled_area_material_removal_structure.

NOTE EXPRESS specification:

```

*)
ENTITY filled_area_material_removal_structure
  ABSTRACT SUPERTYPE OF (ONEOF (electrical_isolation_removal_structure,
    thermal_isolation_removal_structure))
  SUBTYPE OF (material_removal_feature);
  SELF\design_intent_modification.design_intent : conductive_filled_area;
END_ENTITY; -- filled_area_material_removal_structure

( *
```

4.2.386 Flatness_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Flatness_tolerance is a type of Geometric_tolerance_without_referenced_datum_system (see 4.2.423) that specifies or partially specifies the design requirement on the flatness of the Physical_feature_or_part_template (see 4.2.658) to which the Flatness_tolerance is applied.

NOTE Flatness tolerance is defined in terms of tolerance zone definitions in 14.2 of ISO 1101.

Constraints:

WR1: The Shape_element specified by the toleranced_shape_element shall be a Physical_feature_or_part_template.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: The Flatness_tolerance shall not be a Geometric_tolerance_with_applied_material_condition_property.

WR3: The Flatness_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

WR4: If the Flatness_tolerance specifies a Tolerance_zone_definition_with_per_unit_size_specification as the tolerance_zone_or_boundary_definition, there shall be exactly two Length_data_element specified as the per_unit_size by the Tolerance_zone_per_unit_size_specification that is specified as the per_unit_size_specification by the Tolerance_zone_definition_with_per_unit_size_specification.

IP1: The form of the toleranced Shape_element shall be consistent with the form of a shape element for which the application of a flatness tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY flatness_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- flatness_tolerance

( *
```

4.2.387 Font_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Font_definition is a type of Managed_design_object (see 4.2.524). A Font_definition is the definition of a font. Font support shall include outlines using glyphs, as well as font and font family.

NOTE 1 This description is intended to be a complete reference to a font definition. Items to be included are usually font size, aspect ratio, type, etc.

NOTE 2 Font_description supports Part_text_template.

NOTE EXPRESS specification:

```
*)
ENTITY font_definition
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- font_definition

--tt tc1 wd9 added
( *
```

4.2.388 Footprint_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Footprint_definition is a type of Ee_product_definition (see 4.2.323). A Footprint_definition is a mating reference layout item for a specific Package shape. It identifies the reference Package shape, and a reference breakout. The reference breakout is only significant for the metal pattern that is on the top layer. The specific Stratum stackup for the reference breakout is of no significance. The Package is identified by a query on shape and then on the Physical_unit_planar_shape that references a Package.

NOTE It is expected that Vias and Via_dependent_land and Land will be used to support the reference layout, as well as copper shapes to support getting from the mating contact to the breakout terminal. In the case of top surface breakout patterns, the join terminals of the reference layout which are typically Via terminals provide the capability to define the standard connection to the pattern. It is the usual case that there is only layout metal on the top layer.

NOTE EXPRESS specification:

```
*)
ENTITY footprint_definition
  SUBTYPE OF (ee_product_definition);
  reference_breakout : OPTIONAL SET [1:?] OF assembly_group_component;
INVERSE
  padstack : SET [1:?] OF padstack_location_in_footprint_definition FOR footprint;
  shape : footprint_definition_shape FOR shape_characterized_footprint_definition;
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*
    USEDIN(SELF\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') |
      (prpc.name = 'template model')
    )) &#62;= 0;
END_ENTITY; -- footprint_definition

(*
```

4.2.389 Footprint_definition_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Footprint_definition_planar_shape is a type of Managed_design_object (see 4.2.524). A Footprint_definition_planar_shape is the planar shape of a Footprint_definition (see 4.2.388).

The data associated with a Footprint_definition_planar_shape are the following:

- shape_characterized_footprint_definition;
- shape_environment;
- shape_material_condition;
- shape_representation.

4.2.389.1 shape_characterized_footprint_definition

The shape_characterized_footprint_definition specifies the role of the Footprint_definition (see 4.2.388) for the Footprint_definition_planar_shape.

4.2.389.2 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the valid environment for the Footprint_definition_planar_shape. The shape_environment need not be specified for a particular Footprint_definition_planar_shape.

4.2.389.3 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Footprint_definition_planar_shape.

4.2.389.4 shape_representation

The shape_representation specifies the Planar_projected_shape (see 4.2.677) that is the model for the Footprint_definition_planar_shape.

NOTE EXPRESS specification:

```
*)
ENTITY footprint_definition_shape
  SUBTYPE OF (planar_projected_shape);
    shape_characterized_footprint_definition : footprint_definition;
    shape_environment                        : OPTIONAL environment;
    shape_material_condition                : material_condition;
END_ENTITY; -- footprint_definition_shape
(*
```

4.2.390 Footprint_occurrence

A Footprint_occurrence is a type of Assembly_group_component (see 4.2.40). A Footprint_occurrence references a Footprint_definition as a usage_definition. A Footprint_occurrence is not a Physical_component, nor is it a Printed_component, nor is it a Laminate_component. A Footprint_occurrence is specifically provided to eliminate redundant location information for Lands on different layers and at different x,y locations and other layered concepts that are typically grouped to support the placement of a component.

NOTE 1 This Application Object is provided for high density array footprint occurrences.

NOTE 2 This Application Object should not be used to relate Padstack_occurrences. The structure is provided in Footprint_definition to relate a Padstack_definition to a Footprint_definition in a geometric context. Further relationships in the design view will be redundant.

NOTE 3 A Footprint_occurrence_sub_assembly_relationship should be used to related a Land to this Application Object since that relationship contains sufficient information to query the Footprint_definition unambiguously for the Land definition detailed information.

The data associated with a Footprint_occurrence are the following:

— usage_definition.

4.2.390.1 usage_definition

The usage_definition specifies the Footprint_definition (see 4.2.388) for the Footprint_occurrence.

NOTE EXPRESS specification:

```

*)
ENTITY footprint_occurrence
  SUBTYPE OF (assembly_group_component);
  SELF\assembly_component.usage_definition : footprint_definition;
WHERE
  WR1: SIZEOF(USEDIN(SELF,
    'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT')) &#60;= 1;
  WR2: NOT(SIZEOF(USEDIN(SELF,
    'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT')) = 1) OR
    (SIZEOF(QUERY( cmfr &#60;* USEDIN(SELF,
    'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT') |
    'AP210_ARM.ASSEMBLY_GROUP_COMPONENT' IN TYPEOF(cmfr.reusable_component))) = 1);
  WR3: SIZEOF(TYPEOF(SELF) * [
    'AP210_ARM.PRINTED_COMPONENT',
    'AP210_ARM.LAMINATE_COMPONENT',
    'AP210_ARM.PHYSICAL_COMPONENT']) = 0;
END_ENTITY; -- footprint_occurrence
(

```

4.2.391 Footprint_occurrence_sub_assembly_relationship

A Footprint_occurrence_sub_assembly_relationship is a type of Managed_design_object (see 4.2.524). A Footprint_occurrence_sub_assembly_relationship is an association that specifies a subassembly constraint among two or more Land (see 4.2.490) that is aggregated into an Footprint_occurrence (see 4.2.390). A Footprint_occurrence_sub_assembly_relationship does not directly reference the Application Object that is the definition for the Land. Instead, it uses an indexing mechanism provided by terminal_location and sub_assembly_reference_designation to locate the padstack in the Footprint_definition and the land template location in that Padstack_definition.

EXAMPLE 1 A Land may participate in a Footprint_occurrence. In that case, there is no need to populate a separate Assembly_composition_relationship between the Land and the Pcb, since the Assembly_composition_relationship between the Footprint_occurrence and the Pcb establishes the required association.

The data associated with a Footprint_occurrence_sub_assembly_relationship are the following:

- terminal_location;
- sub_assembly_reference_designation;
- related_component;
- relating_component.

4.2.391.1 terminal_location

The terminal_location specifies the string which is the identifier for the Padstack_location_in_footprint which this Footprint_occurrence_sub_assembly_relationship references in the context of the definition of the relating_component.

4.2.391.2 sub_assembly_reference_designation

The sub_assembly_reference_designation specifies the string which is the identifier for the Land in the context of the Padstack_definition specified by the definition of the relating_component.

4.2.391.3 related_component

The related_component specifies the dependent Land (see 4.2.490) for the Footprint_occurrence_sub_assembly_relationship.

4.2.391.4 relating_component

The relating_component specifies the independent Footprint_occurrence (see 4.2.390) which is the aggregation of two or more Assembly_component (see 4.2.34) specified by related_component.

Constraints:

UR1: The combination of relating_component, terminal_location, and sub_assembly_reference_designation shall be unique in a population of Footprint_occurrence_sub_assembly_relationship.

NOTE EXPRESS specification:

```
*)
ENTITY footprint_occurrence_sub_assembly_relationship
  SUBTYPE OF (managed_design_object);
    relating_component : laminate_component;
    relating_component : footprint_occurrence;
    sub_assembly_reference_designation : string;
    terminal_location : string;
UNIQUE
  UR1: sub_assembly_reference_designation,
        relating_component,
        terminal_location;
END_ENTITY; -- footprint_occurrence_sub_assembly_relationship
(*
```

4.2.392 Force_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Force_data_element is a

The data associated with a Force_data_element are the following:

— additional_data.

4.2.392.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY force_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- force_data_element
(*
```


4.2.393 Frequency_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Frequency_data_element is a

The data associated with a Frequency_data_element are the following:

— additional_data.

4.2.393.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY frequency_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- frequency_data_element

(*
```

4.2.394 Frozen_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Frozen_assignment is a type of Managed_design_object (see 4.2.524). A Frozen_assignment is the specification that one or more Frozen_design_object (see 4.2.395) must be unchanged during further design activity.

The data associated with a Frozen_assignment are the following:

— assigned_approval;

— frozen_item.

4.2.394.1 assigned_approval

An assigned_approval specifies the role of the Ee_approval (see 4.2.314) for the Frozen_assignment.

4.2.394.2 frozen_item

The frozen_item specifies a Frozen_design_object (see 4.2.395) for which the design is to remain unchanged in the Frozen_assignment. There shall be one or more Frozen_design_object for a Frozen_assignment.

Constraints:

WR1: Each Frozen_assignment shall not be referenced by any Frozen_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY frozen_assignment
  SUBTYPE OF (managed_design_object);
    frozen_item      : SET [1:?] OF frozen_design_object;
    assigned_approval : ee_approval;
END_ENTITY; -- frozen_assignment

( *
```

4.2.395 Frozen_design_object

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Frozen_design_object is a type of Managed_design_object (see 4.2.524). A Frozen_design_object identifies the Managed_design_object which participates in the Frozen_assignment (see 4.2.394).

NOTE The purpose of Frozen_design_object is for approval of design aspects. The typical use is to freeze allocations between the functional and physical design aspects. This use includes gate swapping and terminal swapping.

Constraints:

IP1: Each member of Frozen_design_object shall not be a member of any of the following: Certification, Contract, Ee_approval, Ee_product_configuration, Ee_product_definition, Ee_product_version, Planned_effectivity, Security_classification, Supplied_product_version, Work_order, or Work_request.

NOTE EXPRESS specification:

```
* )
ENTITY frozen_design_object
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- frozen_design_object

--tt tc1 wd10 modified
(*
```

4.2.396 Functional_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_specification is a type of Managed_design_object (see 4.2.524). A Functional_specification is the information that characterizes the behaviour of a Functional_unit (see 4.2.400) under controlled circumstances using stimulus-response test methods. Detailed properties of the relationship between the reference and characterizing signals define the behaviour.

NOTE The evaluation or categorization of this relationship is not standardized in this document.

The data associated with a Functional_specification are the following:

- associated_definition;
- characteristic_category;
- characterizing_signal;
- reference_signal.

4.2.396.1 associated_definition

The associated_definition specifies the role of a Functional_specification_definition (see 4.2.397) for the Functional_specification. The associated_definition is an aggregation of all the Functional_specification with which it is associated.

4.2.396.2 characteristic_category

The characteristic_category specifies the External_definition (see 4.2.361) of the category for the Functional_specification.

4.2.396.3 characterizing_signal

The characterizing_signal specifies the response Signal_specification (see 4.2.825) for the Functional_specification.

4.2.396.4 reference_signal

The reference_signal specifies the reference or stimulus Signal_specification (see 4.2.825) for the Functional_specification.

NOTE EXPRESS specification:

```

*)
ENTITY functional_specification
  SUBTYPE OF (managed_design_object);
    associated_definition    : functional_specification_definition;
    reference_signal         : signal_specification;
    characterizing_functional_unit : functional_unit;
    characterizing_signal    : signal_specification;
    characteristic_category : external_definition;
WHERE
  WR1: characterizing_functional_unit =
        reference_signal.characterizing_functional_unit;
  WR2: characterizing_functional_unit =
        characterizing_signal.characterizing_functional_unit;
  WR3: characterizing_functional_unit.composed_network =
        associated_definition;
END_ENTITY; -- functional_specification

( *
```

4.2.397 Functional_specification_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_specification_definition is a type of Functional_unit_network_definition (see 4.2.402). A Functional_specification_definition is an aggregation of Functional_specification (see 4.2.396) in addition to the network definition that is used as the measurement context for the Functional_specification.

NOTE EXPRESS specification:

```

*)
ENTITY functional_specification_definition
  SUBTYPE OF (functional_unit_network_definition);
INVERSE
  composing_specification : SET [1:?] OF functional_specification FOR
    associated_definition;
END_ENTITY; -- functional_specification_definition

( *
```

4.2.398 Functional_terminal_group

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_terminal_group is a type of a Managed_design_object (see 4.2.524). A Functional_terminal_group is a group of terminals that have common characteristics and can be acted upon as a group.

The data associated with a Functional_terminal_group are the following:

- group_definition;
- group_type;
- name.

4.2.398.1 group_definition

The group_definition specifies the string that provides the meaning for the Functional_terminal_group.

4.2.398.2 group_type

The group_type specifies the string that is the class for the Functional_terminal_group.

4.2.398.3 name

The name specifies the string that is the identifier for the Functional_terminal_group.

Constraints:

UR1: The name shall be unique within a population of Functional_terminal_group.

NOTE EXPRESS specification:

```
*)
ENTITY functional_terminal_group
  SUBTYPE OF (managed_design_object);
  name          : ee_name;
  group_description : STRING;
  group_type     : STRING;
UNIQUE
  UR1: name;
END_ENTITY; -- functional_terminal_group

( *
```

4.2.399 Functional_terminal_group_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Functional_terminal_group_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Functional_terminal_group_assignment` allocates a `Scalar_terminal_definition` to the `Functional_terminal_group`. This is the mechanism that provides members of the `Functional_terminal_group`.

The data associated with a `Functional_terminal_group_assignment` are the following:

- `composed_group`;
- `functional_usage_view_terminal`.

4.2.399.1 `composed_group`

The `composed_group` specifies the role of the `Functional_terminal_group` for the `Functional_terminal_group_assignment`. The `composed_group` specifies the group that the terminal is a member.

4.2.399.2 `functional_usage_view_terminal`

The `functional_usage_view_terminal` specifies the role of the `Scalar_terminal_definition` for the `Functional_terminal_group_assignment`. The `functional_usage_view_terminal` specifies the terminal that is a member of the group.

NOTE EXPRESS specification:

```
*)
ENTITY functional_terminal_group_assignment
  SUBTYPE OF (managed_design_object);
    composed_group          : functional_terminal_group;
    functional_usage_view_terminal : scalar_terminal_definition;
END_ENTITY; -- functional_terminal_group_assignment

( *
```

4.2.400 `Functional_unit`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Functional_unit` is a type of `Managed_design_object` (see 4.2.524). A `Functional_unit` is an occurrence of a `Functional_unit_definition` (see 4.2.401) in the composition of a `Functional_unit_network_definition` (see 4.2.402).

EXAMPLE 1 Figure 48 is a schematic of a network definition showing hierarchical relationships, as well as the usage view and network definition for a functional block.

The data associated with a Functional_unit are the following:

- access_mechanisms;
- composed_network;
- definition;
- functional_property;
- reference_designation.

4.2.400.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Functional_unit is dependent on the existence of the Functional_unit_terminal (see 4.2.407) that specifies the Functional_unit as its accessed_functional_unit. There shall be one or more Functional_unit_terminal for a Functional_unit.

4.2.400.2 composed_network

The composed_network specifies the role of the Functional_unit_network_definition (see 4.2.402) for the Functional_unit.

4.2.400.3 definition

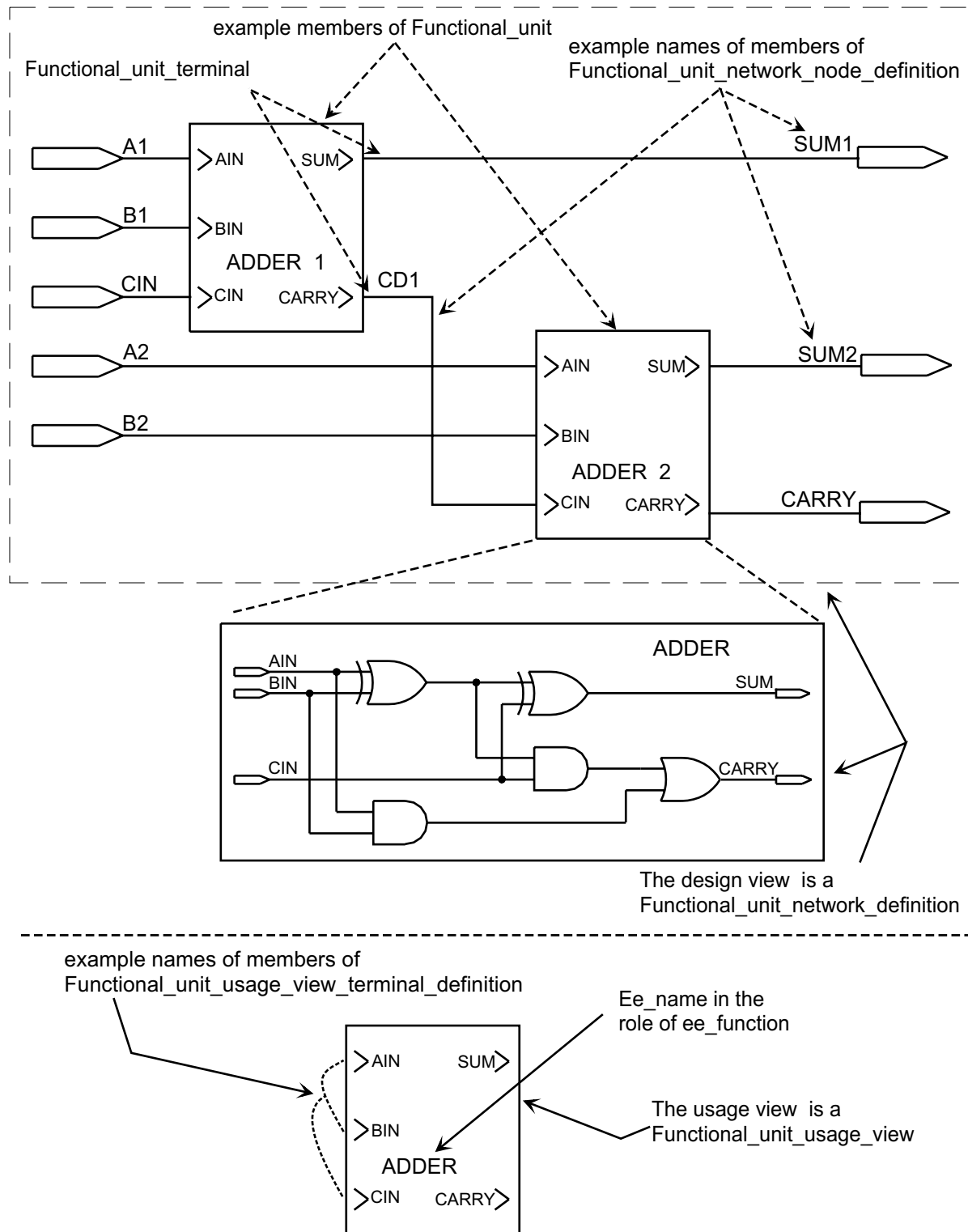
The definition specifies the role of the Functional_unit_definition (see 4.2.401) for the Functional_unit.

4.2.400.4 functional_property

The functional_property specifies the role of the Parameter_assignment (see 4.2.603) for the Functional_unit. There may be more than one Parameter_assignment for the Functional_unit. The functional_property need not be provided for a particular Functional_unit. The functional_property shall be interpreted as overriding the properties supplied as Parameter_assignment for the Functional_unit_definition in the role of definition when the assigned_parameter are identical.

4.2.400.5 reference_designation

The reference_designation specifies the name that identifies the Functional_unit, in the context of the Functional_unit_network_definition (see 4.2.402).



NOTE: This example ignores the difference between the symbol of something, and the thing itself, for presentation purposes.

Figure 48 – Schematic presentation of network definition with Functional_units

Constraints:

UR1: The combination of the reference_designation and the composed_network shall be unique within a population of Functional_units.

WR1: A network definition shall not define itself.

WR2: Considering the functional_unit_definition entities as nodes in a structural graph, and the functional_unit entity as the links of the graph, the graph thus composed shall be directed and acyclic.

NOTE A node that is a Functional_unit_definition may be a Functional_unit_network_definition.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit
  SUBTYPE OF (managed_design_object);
    definition          : functional_unit_definition;
    reference_designation : ee_name;
    composed_network    : functional_unit_network_definition;
    functional_property  : OPTIONAL SET [1:?] OF parameter_assignment;
  INVERSE
    access_mechanisms : SET [1:?] OF functional_unit_terminal FOR
                          accessed_functional_unit;
  UNIQUE
    UR1 : reference_designation, composed_network;
END_ENTITY; -- functional_unit

( *
```

4.2.401 Functional_unit_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_definition is a type of Ee_product_definition (see 4.2.323). A Functional_unit_definition is one of a Functional_unit_network_definition (see 4.2.402), or a Functional_unit_usage_view (see 4.2.410). A Functional_unit_definition is the characterization of behaviour realized by one or more versions of a part or a product, or a subset of a product. This characterization is applicable to every occurrence of the element modelled.

NOTE This part of ISO 10303 explicitly supports requirements traceability from a functional definition to a physical definition by providing application objects with the semantics of a forward design flow (i.e., from functional definition to physical definition). In order to use the standard to exchange functional definitions created only to support analysis of existing physical definitions, it is necessary to populate the discipline_id attribute of the supertype Ee_product_definition. The discipline_id of Ee_product_definition specifies the usage of the Functional_unit_definition. Two possible values are "Analysis", "Synthesis". It may not be possible to correlate each Functional_unit to a corresponding Assembly_component when the value is "Analysis". The value of "Synthesis" is suggested when the purpose is not analysis.

The data associated with a Functional_unit_definition are the following:

— analytical_functional_unit_representation.

4.2.401.1 analytical_functional_unit_representation

The analytical_functional_unit_representation specifies the Analytical_representation (see 4.2.22) for the computer-interpretable model whose contents are not defined in this part of ISO 10303 for the Functional_unit_definition. The analytical_functional_unit_representation need not be specified for a particular Functional_unit_definition. There may be more than one Analytical_representation for an Functional_unit_definition.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_definition
  ABSTRACT SUPERTYPE OF (ONEOF (functional_unit_network_definition,
functional_unit_usage_view))
  SUBTYPE OF (ee_product_definition);
    analytical_functional_unit_representation : OPTIONAL SET [1:?] OF
                                                analytical_representation;
END_ENTITY; -- functional_unit_definition

( *
```

4.2.402 Functional_unit_network_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_network_definition is a type of Functional_unit_definition (see 4.2.401). A Functional_unit_network_definition may be either one or more of an Electrical_network_definition (see 4.2.345) or a Thermal_network_definition (see 4.2.906) or a Functional_specification_definition (see 4.2.397).

The data associated with a Functional_unit_network_definition are the following:

— usage_view.

4.2.402.1 usage_view

The usage_view specifies the role of the Functional_unit_usage_view (see 4.2.410) for the Functional_unit_network_definition.

Constraints:

WR1: The member of Functional_unit_usage_view specified as a usage_view by this member of functional_unit_network_definition shall be an Ee_product_definition of the same version of the product for which this member of Functional_unit_network_definition is an Ee_product_definition.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_network_definition
  SUPERTYPE OF (thermal_network_definition ANDOR
    electrical_network_definition ANDOR functional_specification_definition)
  SUBTYPE OF (functional_unit_definition);
  usage_view : functional_unit_usage_view;
END_ENTITY; -- functional_unit_network_definition

( *
```

4.2.403 Functional_unit_network_node_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_network_node_definition is a type of Managed_design_object (see 4.2.524). A Functional_unit_network_node_definition is the definition of a node in the Functional_unit_network_definition (see 4.2.402).

The data associated with a Functional_unit_network_node_definition are the following:

- associated_functional_unit_definition;
- external_node_access;
- functional_unit_network_node_name.

4.2.403.1 associated_functional_unit_definition

The associated_functional_unit_definition specifies the Functional_unit_network_definition (see 4.2.402) that the Functional_unit_network_node_definition is helping to compose.

4.2.403.2 external_node_access

The external_node_access is an inverse relationship that specifies the Functional_unit_network_terminal_definition_node_assignment (see 4.2.405) that specifies the Functional_unit_network_node_definition as its composed_node. There shall be no more than one Functional_unit_network_terminal_definition_node_assignment for a Functional_unit_network_node_definition.

4.2.403.3 functional_unit_network_node_name

The functional_unit_network_node_name identifies the Functional_unit_network_node_definition in the context of the associated_functional_unit_definition.

Constraints:

UR1: The combination of the functional_unit_network_node_name and the associated_functional_unit_definition shall be unique within a population of Functional_unit_network_node_definitions.

WR1: A member of Functional_unit_network_node_definition shall be referenced by no more than one member of Functional_unit_network_terminal_definition_node_assignment.

IP1: A member of Functional_unit_network_node_definition that is helping to compose an approved electrical or thermal network definition shall be referenced by at least one of the following: a member of Bus_element_link; a member of Functional_unit_terminal_node_assignment.

IP2: A member of Functional_unit_network_node_definition that is helping to compose an approved electrical or thermal network definition shall be referenced by at least two of the following: a member of Bus_element_link; a member of Functional_unit_terminal_node_assignment; no more than one member of Functional_unit_network_terminal_definition_node_assignment.

IP3: Each member of Functional_unit_network_node_definition shall be referenced by no more than one member of Aggregate_connectivity_requirement for a specific physical partition.

NOTE Some nodes are not connected to external terminals.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_network_node_definition
  SUBTYPE OF (managed_design_object);
    associated_functional_unit_definition : functional_unit_network_definition;
    functional_unit_network_node_name      : ee_name;
DERIVE
  valid_conservative_node : BOOLEAN := ((SIZEOF(QUERY( vc &#60;* USEDIN (SELF,
    'AP210_ARM.BUS_ELEMENT_LINK.PRECEDENT_ELEMENT')|
    vc &#60;&#62; [ ])) +
    SIZEOF(QUERY( vc &#60;* USEDIN (SELF,
    'AP210_ARM.BUS_ELEMENT_LINK.SUBSEQUENT_ELEMENT')|
    vc &#60;&#62; [ ])) +
    SIZEOF(QUERY( vc &#60;* USEDIN (SELF,
    'AP210_ARM.FUNCTIONAL_UNIT_TERMINAL_NODE_ASSIGNMENT.COMPOSED_NODE') |
    vc &#60;&#62; [ ])) +
    SIZEOF(external_node_access)) &#62; 1);

INVERSE
  external_node_access : SET [0:1] OF
    functional_unit_network_terminal_definition_node_assignment FOR
    composed_node;

UNIQUE
  UR1 : functional_unit_network_node_name, associated_functional_unit_definition;
END_ENTITY; -- functional_unit_network_node_definition

( *
```

4.2.404 Functional_unit_network_terminal_definition_bus_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

Functional_unit_network_terminal_definition_bus_assignment is a type of Managed_design_object (see 4.2.524). A Functional_unit_network_terminal_definition_bus_assignment is the relationship between the structural definition of a bus and the terminal definition of a network to which the bus is assigned.

The data associated with a Functional_unit_network_terminal_definition_bus_assignment are the following:

- connected_bus;
- connected_terminal.

4.2.404.1 connected_bus

The connected_bus specifies the role of the Bus_structural_definition (see 4.2.83) for the Functional_unit_network_terminal_definition_bus_assignment.

4.2.404.2 connected_terminal

The `connected_terminal` specifies the role of the `Functional_unit_usage_view_terminal_definition` (see 4.2.411) for the `Functional_unit_network_terminal_definition_bus_assignment`.

NOTE EXPRESS specification:

```
*)
ENTITY functional_unit_network_terminal_definition_bus_assignment
  SUBTYPE OF (managed_design_object);
    connected_bus      : bus_structural_definition;
    connected_terminal : functional_unit_usage_view_terminal_definition;
END_ENTITY; -- functional_unit_network_terminal_definition_bus_assignment

( *
```

4.2.405 Functional_unit_network_terminal_definition_node_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Functional_unit_network_terminal_definition_node_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Functional_unit_network_terminal_definition_node_assignment` is the assignment relationship between a node definition of the functional unit network definition and a terminal definition of that network definition. This assignment provides visibility into the node definition from higher levels in the functional decomposition.

The data associated with a `Functional_unit_network_terminal_definition_node_assignment` are the following:

- `composed_node`;
- `connected_terminal`.

4.2.405.1 composed_node

The `composed_node` specifies the role of the `Functional_unit_network_node_definition` (see 4.2.403) for the `Functional_unit_network_terminal_definition_node_assignment`.

4.2.405.2 connected_terminal

The `connected_terminal` specifies the role of the `Functional_unit_usage_view_terminal_definition` (see 4.2.411) for the `Functional_unit_network_terminal_definition_node_assignment`.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_network_terminal_definition_node_assignment
  SUBTYPE OF (managed_design_object);
    composed_node      : functional_unit_network_node_definition;
    connected_terminal : functional_unit_usage_view_terminal_definition;
END_ENTITY; -- functional_unit_network_terminal_definition_node_assignment

( *
```

4.2.406 Functional_unit_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_requirement_allocation is a type of Managed_design_object (see 4.2.524). A Functional_unit_requirement_allocation is the relationship between the Ee_requirement_occurrence (see 4.2.330) and the Functional_unit (see 4.2.400) that satisfies the Ee_requirement_occurrence.

The data associated with a Functional_unit_requirement_allocation are the following:

- functional_requirement;
- functional_unit_assignment.

4.2.406.1 functional_requirement

The functional_requirement specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Functional_unit_requirement_allocation.

4.2.406.2 functional_unit_assignment

The functional_unit_assignment specifies the Functional_unit (see 4.2.400) that satisfies the Ee_requirement_occurrence (see 4.2.330) for the Functional_unit_requirement_allocation.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_requirement_allocation
  SUBTYPE OF (managed_design_object);
    functional_requirement : ee_requirement_occurrence;
    functional_unit_assignment : functional_unit;
END_ENTITY; -- functional_unit_requirement_allocation

( *
```

4.2.407 Functional_unit_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_terminal is a type of Managed_design_object (see 4.2.524). A Functional_unit_terminal is an access mechanism for a Functional_unit (see 4.2.400).

The data associated with a Functional_unit_terminal are the following:

- accessed_functional_unit;
- bus_assignment;
- definition;
- node_assignment.

4.2.407.1 accessed_functional_unit

The accessed_functional_unit specifies the Functional_unit (see 4.2.400) whose functionality is accessed by the Functional_unit_terminal.

4.2.407.2 bus_assignment

The bus_assignment is an inverse relationship that specifies the Functional_unit_terminal_bus_assignment (see 4.2.408) that specifies the Functional_unit_terminal as its connected_terminal. There shall be no more than one Functional_unit_terminal_bus_assignment for a Functional_unit_terminal.

4.2.407.3 definition

The definition specifies the role of the Functional_unit_usage_view_terminal_definition (see 4.2.411) for the Functional_unit_terminal.

4.2.407.4 node_assignment

The node_assignment is an inverse relationship that specifies the Functional_unit_terminal_node_assignment (see 4.2.409) that specifies the Functional_unit_terminal as its connected_terminal. There shall be no more than one Functional_unit_terminal_node_assignment for a Functional_unit_terminal.

Constraints:

WR1: The definition of the Functional_unit this Functional_unit_terminal is associated with shall be either a Functional_unit_network_definition or a Functional_unit_usage_view.

WR2: Each Functional_unit_terminal shall not be assigned to a bus and to a node simultaneously.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_terminal
  SUBTYPE OF (managed_design_object);
    accessed_functional_unit : functional_unit;
    definition                : functional_unit_usage_view_terminal_definition;
  INVERSE
    bus_assignment  : SET [0:1] OF
                      functional_unit_terminal_bus_assignment FOR
                      connected_terminal;
    node_assignment : SET [0:1] OF
                      functional_unit_terminal_node_assignment FOR
                      connected_terminal;
END_ENTITY; -- functional_unit_terminal

( *

```

4.2.408 Functional_unit_terminal_bus_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_terminal_bus_assignment is a type of Managed_design_object (see 4.2.524). A Functional_unit_terminal_bus_assignment is the relationship between the structural definition of a bus and a terminal that is assigned to the bus. The members of Functional_unit_terminal (see 4.2.407) that are assigned to the same bus by this application object are defined to be connected to each other.

NOTE There are cardinality constraints levied to ensure that the cardinality as resolved through all the references is identical.

The data associated with a Functional_unit_terminal_bus_assignment are the following:

- connected_bus;
- connected_terminal.

4.2.408.1 connected_bus

The connected_bus specifies the role of the Bus_structural_definition (see 4.2.83) for the Functional_unit_terminal_bus_assignment.

4.2.408.2 connected_terminal

The connected_terminal specifies the role of the Functional_unit_terminal (see 4.2.407) for the Functional_unit_terminal_bus_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_terminal_bus_assignment
  SUBTYPE OF (managed_design_object);
    connected_bus      : bus_structural_definition;
    connected_terminal : functional_unit_terminal;
END_ENTITY; -- functional_unit_terminal_bus_assignment

( *
```

4.2.409 Functional_unit_terminal_node_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_terminal_node_assignment is a type of Managed_design_object (see 4.2.524). A Functional_unit_terminal_node_assignment is the assignment relationship between a Functional_unit_network_node_definition (see 4.2.403) and a Functional_unit_terminal (see 4.2.407).

The data associated with a Functional_unit_terminal_node_assignment are the following:

- composed_node;
- connected_terminal.

4.2.409.1 composed_node

The composed_node specifies the role of the Functional_unit_network_node_definition (see 4.2.403) for the Functional_unit_terminal_node_assignment.

4.2.409.2 connected_terminal

The connected_terminal specifies the role of the Functional_unit_terminal (see 4.2.407) for the Functional_unit_terminal_node_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_terminal_node_assignment
  SUBTYPE OF (managed_design_object);
    composed_node      : functional_unit_network_node_definition;
    connected_terminal : functional_unit_terminal;
END_ENTITY; -- functional_unit_terminal_node_assignment

( *
```

4.2.410 Functional_unit_usage_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_usage_view is a type of Functional_unit_definition (see 4.2.401) that describes the usage of the Functional_unit (see 4.2.400) in a Functional_unit_definition.

NOTE In the case where the Functional_unit_usage_view is an Ee_product_definition for a 'functionality' product, the name of the function being represented by this Functional_unit_usage_view is represented by the Ee_product.nomenclature. In the case where the Functional_unit_usage_view is an Ee_product_definition for a physical product, e.g., a netlist, there is no need for a separate name for the netlist since it is identified by the product_definition_id inherited from Ee_product_definition. In fact, product_definition_id always identifies the Functional_unit_usage_view.

The data associated with a Functional_unit_usage_view are the following:

— access_mechanism.

4.2.410.1 access_mechanism

The access_mechanism is an inverse relationship that specifies that the existence of the Functional_unit_usage_view is dependent on the existence of the Functional_unit_usage_view_terminal_definition (see 4.2.411) that specifies the Functional_unit_usage_view as its associated_functional_unit_definition. There shall be one or more Functional_unit_usage_view_terminal_definition for a Functional_unit_usage_view.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_usage_view
  SUBTYPE OF (functional_unit_definition);
  INVERSE
    access_mechanism : SET [1:?] OF functional_unit_usage_view_terminal_definition FOR
                        associated_functional_unit_definition;
END_ENTITY; -- functional_unit_usage_view

( *
```

4.2.411 Functional_unit_usage_view_terminal_definition

A Functional_unit_usage_view_terminal_definition is a type of Managed_design_object (see 4.2.524). A Functional_unit_usage_view_terminal_definition may be a Scalar_terminal_definition (see 4.2.799). A Functional_unit_usage_view_terminal_definition is the definition of a Functional_unit_terminal (see 4.2.407).

The data associated with a Functional_unit_usage_view_terminal_definition are the following:

- associated_functional_unit_definition;
- bus_assignment;
- node_assignment;
- signal_name.

4.2.411.1 associated_functional_unit_definition

The associated_functional_unit_definition specifies the role of the Functional_unit_usage_view (see 4.2.410) for the Functional_unit_usage_view_terminal_definition.

4.2.411.2 bus_assignment

The bus_assignment is an inverse relationship that specifies the Functional_unit_network_terminal_definition_bus_assignment (see 4.2.404) that specifies the Functional_unit_terminal (see 4.2.407) as its connected_terminal. There shall be no more than one Functional_unit_network_terminal_definition_bus_assignment for a Functional_unit_usage_view_terminal_definition.

4.2.411.3 node_assignment

The node_assignment is an inverse relationship that specifies the Functional_unit_network_terminal_definition_node_assignment (see 4.2.405) that specifies the Functional_unit_terminal (see 4.2.407) as its connected_terminal. There shall be no more than one Functional_unit_network_terminal_definition_node_assignment for a Functional_unit_usage_view_terminal_definition.

4.2.411.4 signal_name

The signal_name identifies the Functional_unit_usage_view_terminal_definition in the context of the Functional_unit_usage_view (see 4.2.410).

Constraints:

UR1: The combination of the signal_name and the associated_functional_unit_definition shall be unique within a population of Functional_unit_usage_view_terminal_definitions.

WR1: Each Functional_unit_usage_view_terminal_definition shall not be assigned to a bus and to a node simultaneously.

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_usage_view_terminal_definition
  SUPERTYPE OF (scalar_terminal_definition)
  SUBTYPE OF (managed_design_object);
  associated_functional_unit_definition : functional_unit_usage_view;
  signal_name                          : ee_name;
  INVERSE
    node_assignment : SET [0:?] OF
                        functional_unit_network_terminal_definition_node_assignment FOR
                        connected_terminal;
    bus_assignment  : SET [0:?] OF
                        functional_unit_network_terminal_definition_bus_assignment FOR
                        connected_terminal;
  UNIQUE
    UR1 : signal_name, associated_functional_unit_definition;
  WHERE
    WR1: (EXISTS(node_assignment) AND NOT(EXISTS(bus_assignment))) OR
          (NOT(EXISTS(node_assignment)) AND EXISTS(bus_assignment)) OR
          (NOT(EXISTS(node_assignment)) AND NOT(EXISTS(bus_assignment)));
END_ENTITY; -- functional_unit_usage_view_terminal_definition
( *

```

4.2.412 Functional_unit_usage_view_terminal_definition_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_unit_usage_view_terminal_definition_make_from_relationship is a type of Managed_design_object (see 4.2.524).

NOTE EXPRESS specification:

```

*)
ENTITY functional_unit_usage_view_terminal_definition_make_from_relationship
  SUBTYPE OF (managed_design_object);
  associated_make_from : engineering_make_from;
  reusable_feature    : functional_unit_usage_view_terminal_definition;
  resultant_feature   : functional_unit_usage_view_terminal_definition;
  UNIQUE
    UR1: reusable_feature, resultant_feature;
  WHERE
    WR1: reusable_feature :&#60;&#62;; resultant_feature;
    WR2: reusable_feature.associated_functional_unit_definition :&#60;&#62;;
          resultant_feature.associated_functional_unit_definition;
END_ENTITY; -- functional_unit_usage_view_terminal_definition_make_from_relationship
( *

```

4.2.413 Functional_usage_view_to_part_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_usage_view_to_part_terminal_assignment is a type of Managed_design_object (see 4.2.524). A Functional_usage_view_to_part_terminal_assignment is the association between a Scalar_terminal_definition (see 4.2.799) and a Part_terminal (see 4.2.630). This is the fundamental mechanism for mapping between the functional view and the physical view of a design or library item.

The data associated with a Functional_usage_view_to_part_terminal_assignment are the following:

- functional_usage_view_terminal;
- physical_usage_view_terminal.

4.2.413.1 functional_usage_view_terminal

The functional_usage_view_terminal specifies the role of the Scalar_terminal_definition (see 4.2.799) for the Functional_usage_view_to_part_terminal_assignment.

4.2.413.2 physical_usage_view_terminal

The physical_usage_view_terminal specifies the role of the Part_terminal (see 4.2.630) for the Functional_usage_view_to_part_terminal_assignment.

Constraints:

UR1: The combination of functional_usage_view_terminal and physical_usage_view_terminal shall be unique within a population of Functional_usage_view_to_part_terminal_assignment.

WR1: The members of functional_usage_view_to_part_terminal_assignment that specify the same member of physical_unit_usage_view through the path physical_usage_view_terminal.associated_definition shall all specify the same member of functional_unit_definition through the path functional_usage_view_terminal.associated_functional_unit_definition.

NOTE EXPRESS specification:

```

*)
ENTITY functional_usage_view_to_part_terminal_assignment
  SUBTYPE OF (managed_design_object);
    physical_usage_view_terminal : part_terminal;
    functional_usage_view_terminal : scalar_terminal_definition;
  UNIQUE
    UR1 : functional_usage_view_terminal, physical_usage_view_terminal;
END_ENTITY; -- functional_usage_view_to_part_terminal_assignment

( *

```

4.2.414 Functional_usage_view_to_printed_part_template_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Functional_usage_view_to_printed_part_template_terminal_assignment is a type of Managed_design_object (see 4.2.524). A Functional_usage_view_to_printed_part_template_terminal_assignment is the association between a functional terminal definition and a terminal of a Printed_part_template (see 4.2.730).

EXAMPLE 1 It is permissible for one functional_usage_view_terminal to be assigned to more than one physical_usage_view_terminal, for example, VCC.

The data associated with a Functional_usage_view_to_printed_part_template_terminal_assignment are the following:

- functional_usage_view_terminal;
- physical_usage_view_terminal.

4.2.414.1 functional_usage_view_terminal

The functional_usage_view_terminal specifies the role of the Scalar_terminal_definition (see 4.2.799) for the Functional_usage_view_to_printed_part_template_terminal_assignment.

4.2.414.2 physical_usage_view_terminal

The physical_usage_view_terminal specifies the role of the Printed_part_template_terminal (see 4.2.737) for the Functional_usage_view_to_printed_part_template_terminal_assignment.

Constraints:

UR1: The combination of functional_usage_view_terminal and physical_usage_view_terminal shall be unique within a population of Functional_usage_view_to_printed_part_template_terminal_assignment.

WR1: The members of functional_usage_view_to_printed_part_template_terminal_assignment that specify the same member of physical_unit_usage_view through the path physical_usage_view_terminal.associated_definition shall all specify the same member of functional_unit_definition through the path functional_usage_view_terminal.associated_functional_unit_definition.

NOTE EXPRESS specification:

```

*)
ENTITY functional_usage_view_to_printed_part_template_terminal_assignment
  SUBTYPE OF (managed_design_object);
    physical_usage_view_terminal : printed_part_template_terminal;
    functional_usage_view_terminal : scalar_terminal_definition;
  UNIQUE
    UR1 : functional_usage_view_terminal, physical_usage_view_terminal;
END_ENTITY; -- functional_usage_view_to_printed_part_template_terminal_assignment

( *
```

4.2.415 Gate_path_swap_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Gate_path_swap_specification is a type of Change_design_object_management_relationship (see 4.2.95). This entity explicitly states that a path to a functional unit in a design shall be changed to a new path in the library of functional unit definitions.

NOTE Normally the new path is based on the same root functional unit definition in the library, leading to a different path in the same technology, but there is no restriction in the standard that prohibits switching to a different functional unit definition in the library. If there is a new functional unit definition as root, other design changes will likely need to occur, but the mapping to identify those explicitly and relate them to this entity is not standardized, except through the use of the requirement_occurrence concept. Specifically, this entity needs to be used in conjunction with the component swap specification to swap between components if the path in the library is also changing.

EXAMPLE 1 An example is in the case where a change is being made from one microcircuit and its terminal, U1-A, to another microcircuit and terminal, U2-C.

The data associated with a Gate_path_swap_specification are the following:

- current_design_object;
- previous_design_object.

4.2.415.1 current_design_object

The `current_design_object` specifies the `Design_functional_unit_allocation_to_reference_functional_unit` (see 4.2.266) that results from the swap in the `Gate_path_swap_specification`.

4.2.415.2 previous_design_object

The `previous_design_object` specifies the `Design_functional_unit_allocation_to_reference_functional_unit` (see 4.2.266) from which the swap was made in the `Gate_path_swap_specification`.

NOTE EXPRESS specification:

```

*)
ENTITY gate_path_swap_specification
  SUBTYPE OF (change_design_object_management_relationship);
    SELF\change_design_object_management_relationship.current_design_object :
design_functional_unit_allocation_to_reference_functional_unit;
    SELF\change_design_object_management_relationship.previous_design_object :
design_functional_unit_allocation_to_reference_functional_unit;
END_ENTITY; -- gate_path_swap_specification

( *
```

4.2.416 Geometric_alignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Geometric_alignment` is a type of `Derived_shape_element` (see 4.2.260) that is a linear or planar `Shape_element` (see 4.2.810) that is established to contain two or more aligned `Shape_element`. The `Shape_element` from which the `Geometric_alignment` is established are associated with the `Geometric_alignment` by the `Shape_element_deriving_relationship` (see 4.2.814) that are specified as the deriving-relationships.

NOTE The attribute `deriving_relationships` is declared in the `Derived_shape_element` entity of which this entity is a subtype.

EXAMPLE 1 A `Geometric_alignment` that is derived from the `Centre_axis` of a circular hole pattern and a `Centre_axis` of one of the holes of the pattern is a plane to which a `Dimensional_location` might be applied.

Constraints:

WR1: The `Geometric_alignment` shall be derived from two or more `Shape_element`.

IP1: In the case of a planar `Derived_shape_element`, at least one of the aligned `Shape_element` shall be a linear `Shape_element`.

NOTE EXPRESS specification:

```
*)
ENTITY geometric_alignment
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- geometric_alignment

( *
```

4.2.417 Geometric_intersection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Geometric_intersection is a type of Derived_shape_element (see 4.2.260) that is the common intersection of two or more Shape_element (see 4.2.810). The Shape_element from which the Geometric_intersection is derived are associated with the Geometric_intersection by the Shape_element_deriving_relationship (see 4.2.814) that are specified as the deriving_relationships.

NOTE The attribute deriving_relationships is declared in the Derived_shape_element entity of which this entity is a subtype.

Constraints:

WR1: The Geometric_intersection shall be derived from two or more Shape_element.

IP1: The Shape_element from which the Geometric_intersection is derived shall intersect.

NOTE EXPRESS specification:

```
*)
ENTITY geometric_intersection
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- geometric_intersection

( *
```

4.2.418 Geometric_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Geometric_representation_relationship is a type of Managed_design_object (see 4.2.524). A Geometric_representation_relationship may be a Shape_composition_relationship (see 4.2.807), or a Component_part_2d_geometric_relationship or a Component_part_3d_relationship or a Component_part_non_planar_relationship. A Geometric_representation_relationship is the association between a Shape_definition (see 4.2.808) and another Shape_definition (see 4.2.808) for predefined purposes.

NOTE The purposes are described in the subtypes of this Application Object.

The data associated with a Geometric_representation_relationship are the following:

- shape_1;
- shape_2.

4.2.418.1 shape_1

The shape_1 specifies one role of the Shape_definition (see 4.2.808) for the Geometric_representation_relationship.

4.2.418.2 shape_2

The shape_1 specifies one role of the Shape_definition (see 4.2.808) for the Geometric_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY geometric_representation_relationship
  SUPERTYPE OF (ONEOF(
    shape_composition_relationship,
    component_part_2d_geometric_representation_relationship,
    component_part_2d_non_planar_geometric_representation_relationship,
    component_part_3d_geometric_representation_relationship))
  SUBTYPE OF (managed_design_object);
  shape_1 : shape_definition;
  shape_2 : shape_definition;
WHERE
  WR1: shape_1 &#60;&#62; shape_2;
--acyclic relationship
END_ENTITY; -- geometric_representation_relationship
(

```

4.2.419 Geometric_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Geometric_tolerance` is a type of `Managed_design_object` (see 4.2.524). A `Geometric_tolerance` is one of a `Geometric_tolerance_without_referenced_datum_system` (see 4.2.423) or a `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422). Additionally a `Geometric_tolerance` may be one of a `Position_tolerance` (see 4.2.695), a `Profile_of_any_line_tolerance` (see 4.2.744), or a `Profile_of_any_surface_tolerance` (see 4.2.747). A `Geometric_tolerance` may be `Geometric_tolerance_with_applied_material_condition_property` (see 4.2.421). Additionally, a `Geometric_tolerance` may be a `Statistical_geometric_tolerance` (see 4.2.841).

A `Geometric_tolerance` is a specification or partial specification of a design requirement on one or more of the form, location, or orientation characteristics of a `Shape_element` (see 4.2.810). This type of design requirement is specified in terms of a `Tolerance_zone` (see 4.2.913) within which the toleranced `Shape_element` must reside; or in the case of a `Boundary_based_position_tolerance` (see 4.2.77), a boundary that the toleranced `Shape_element` must not violate.

A `Geometric_tolerance_with_referenced_datum_system` participates in specifying a design requirement when the requirement is on one or more characteristics such as location of the toleranced `Shape_element` in relation to a `Datum_system` (see 4.2.236). In such cases, the `Tolerance_zone` is located and/or oriented, and thus the toleranced `Shape_element` is located and/or oriented, in relation to the `Datum_system`.

In some cases the design requirement is such that the boundary of the `Tolerance_zone` is on a per unit basis.

EXAMPLE 1 The design requirement on the flatness of a surface is such that the surface should not deviate by more than 0.5 mm in any 10 mm by 10 mm area. A `Geometric_tolerance` that defines a `Tolerance_zone_definition_with_per_unit_size_specification` (see 4.2.921) participates in specifying this type of requirement.

In some cases a design requirement on the geometric characteristic or characteristics of a `Shape_element` is such that it is dependent upon the produced size of the `Shape_element`, or upon the produced size of the `Shape_element` from which it is derived. A `Geometric_tolerance_with_applied_material_condition_property` specifies this type of requirement.

The data associated with a `Geometric_tolerance` are the following:

- `tolerance_zone_or_boundary_definition`;
- `toleranced_shape_element`.

4.2.419.1 tolerance_zone_or_boundary_definition

The `tolerance_zone_or_boundary_definition` specifies either a `Conical_tolerance_zone_boundary_diametrical_size_characteristic`, a `Positional_boundary_member_definition` (see 4.2.702), a `Profile_related_positional_boundary_definition` (see 4.2.751), or a `Tolerance_zone_definition` (see 4.2.920) that defines the constraints placed on the `Shape_element` (see 4.2.810) specified as the `toleranced_shape_element`.

4.2.419.2 toleranced_shape_element

The `toleranced_shape_element` specifies either a `Centre_Axis`, `Centre_plane` (see 4.2.92), `Centre_point` (see 4.2.93), `Physical_feature_or_part_template` (see 4.2.658), or `Tangent_plane` (see 4.2.884) to which the `Geometric_tolerance` is applied.

Constraints:

WR1: The `Geometric_tolerance` shall either be a `Geometric_tolerance_with_referenced_datum_system` or a `Geometric_tolerance_without_referenced_datum_system`.

WR2: The `Geometric_tolerance` shall be specified by at most one `Simultaneous_geometric_tolerance_-requirement` as one of the `grouped_geometric_tolerances`.

WR3: There shall be exactly one `Ee_requirement_occurrence` with the description of "Geometric Dimensioning and Tolerancing Reference" that specifies at least one `Ee_specification` as the `requirement_specification`.

IP1: There shall be exactly one `Ee_requirement_occurrence` with the description of "Geometric Dimensioning and Tolerancing Reference" that specifies at least one `Ee_specification` as the `requirement_specification`. The `Ee_specification` shall reference those standards by which the geometric dimensioning and tolerancing data is to be interpreted.

EXAMPLE 1 An `Ee_requirement_occurrence` with a description of "Geometric Dimensioning and Tolerancing Reference" specifies an `Ee_specification` as the `requirement_specification`. The `Ee_specification` may specify "Technical drawings – Geometrical tolerancing – Tolerancing of form, orientation, location and run-out – Generalities, definitions, symbols, indications on drawings" as the `document_name`, 1995 as the `year_of_the_creation_date`, "ISO" as the `publisher`, and an `Ee_document_identifier` as the `base_identifier` that has an `assigned_value` of "1101" and an `identifier_name` of "document number".

NOTE EXPRESS specification:

```
*)
ENTITY geometric_tolerance
  ABSTRACT SUPERTYPE OF (((ONEOF (
    geometric_tolerance_with_referenced_datum_system,
    geometric_tolerance_without_referenced_datum_system) ANDOR ONEOF (
    profile_of_any_line_tolerance, position_tolerance,
    profile_of_any_surface_tolerance)) ANDOR
    geometric_tolerance_with_applied_material_condition_property) ANDOR
    statistical_geometric_tolerance)
  SUBTYPE OF (managed_design_object);
    tolerance_zone_or_boundary_definition : constraining_element_definitions;
    toleranced_shape_element              : toleranceable_shape_element;
END_ENTITY; -- geometric_tolerance

(*
```

4.2.420 Geometric_tolerance_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Geometric_tolerance_qualifier is a type of Managed_design_object (see 4.2.524). A Geometric_tolerance_qualifier is either a Pre_defined_geometric_tolerance_qualifier (see 4.2.708) or a User_defined_geometric_tolerance_qualifier (see 4.2.945). A Geometric_tolerance_qualifier is a description that qualifies one or more Geometric_tolerance (see 4.2.419).

NOTE Qualifying notes are described in 5.3 of ISO/DIS 1101.

The data associated with a Geometric_tolerance_qualifier are the following:

— qualified_geometric_tolerances.

4.2.420.1 qualified_geometric_tolerances

The qualified_geometric_tolerances specifies the Geometric_tolerance (see 4.2.419) to which the Geometric_tolerance_qualifier applies. There shall be one or more Geometric_tolerance for a Geometric_tolerance_qualifier.

NOTE EXPRESS specification:

```
*)
ENTITY geometric_tolerance_qualifier
  ABSTRACT SUPERTYPE OF (ONEOF (
    pre_defined_geometric_tolerance_qualifier,
    user_defined_geometric_tolerance_qualifier))
  SUBTYPE OF (managed_design_object);
  qualified_geometric_tolerances : SET [1:?] OF geometric_tolerance;
END_ENTITY; -- geometric_tolerance_qualifier

(*
```

4.2.421 Geometric_tolerance_with_applied_material_condition_property

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Geometric_tolerance_with_applied_material_condition_property` is a type of `Geometric_tolerance` (see 4.2.419) that specifies or partially specifies the design requirement on one or more geometric characteristics of the toleranced `Shape_element` (see 4.2.810) in which the restrictiveness of the requirement is dependent on the actual size of the produced `Shape_element` or the actual size of the produced `Shape_element` from which the toleranced `Shape_element` is derived. A `Geometric_tolerance_with_applied_material_condition_property` in which the `applied_material_condition_property_name` is "maximum_material_principle" participates in specifying a requirement that is less stringent as the produced `Shape_element` deviates from its maximum material condition towards its least material condition. In this case, the specified size of the `Tolerance_zone_boundary` (see 4.2.915) is the base size from which it is increased by the amount the produced `Shape_element` deviates from its maximum material condition towards its least material condition. A `Geometric_tolerance_with_applied_material_condition_property` in which the `applied_material_condition_property_name` is "least_material_requirement" participates in specifying a requirement that is less stringent as the produced `Shape_element` deviates from its least material condition towards its maximum material condition. In this case the specified size of the `Tolerance_zone_boundary` is the base size from which it is increased by the amount the produced `Shape_element` deviates from its least material condition towards its maximum material condition. A `Geometric_tolerance_with_applied_material_condition_property` may be a `Boundary_based_position_tolerance` (see 4.2.77).

The data associated with a `Geometric_tolerance_with_applied_material_condition_property` are the following:

— `applied_material_condition_property_name`.

4.2.421.1 `applied_material_condition_property_name`

The `applied_material_condition_property_name` specifies the type of the applied material condition property, either "least_material_requirement" or "maximum_material_principle" for the `Geometric_tolerance_with_applied_material_condition_property`.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` shall either be a `Centre_axis`, `Centre_plane`, `Centre_point`, or `Physical_feature_or_part_template`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

WR2: If the `Shape_element` specified as the `toleranced_shape_element` is a `Physical_feature_or_part_template`, the `Geometric_tolerance_with_applied_material_condition_property` shall also be a `Boundary_based_position_tolerance`.

NOTE EXPRESS specification:

```

*)
ENTITY geometric_tolerance_with_applied_material_condition_property
  SUPERTYPE OF (boundary_based_position_tolerance)
  SUBTYPE OF (geometric_tolerance);
  applied_material_condition_property_name : material_condition_property_names;
END_ENTITY; -- geometric_tolerance_with_applied_material_condition_property

( *

```

4.2.422 Geometric_tolerance_with_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Geometric_tolerance_with_referenced_datum_system` is a type of `Geometric_tolerance` (see 4.2.419) that specifies or partially specifies the design requirement on one or more geometric characteristics, such as orientation, of the toleranced `Shape_element` (see 4.2.810) in relation to the `Datum_system` (see 4.2.236) defined by the `Datum_system_defining_conditions` (see 4.2.237) that is specified as the `datum_system_definition`. In such cases, the `Tolerance_zone` (see 4.2.913) is located and/or oriented; and thus the toleranced `Shape_element` is located and/or oriented; in relation to the `Datum_system`. A `Geometric_tolerance_with_referenced_datum_system` is one of an `Angularity_tolerance` (see 4.2.26), a `Circular_runout_tolerance` (see 4.2.102), a `Concentricity_and_coaxiality_tolerance` (see 4.2.168), a `Parallelism_tolerance` (see 4.2.602), a `Perpendicularity_tolerance` (see 4.2.645), a `Profile_of_any_line_tolerance_with_referenced_datum_system` (see 4.2.745), a `Profile_of_any_surface_tolerance_with_referenced_datum_system` (see 4.2.748), a `Position_tolerance_with_referenced_datum_system` (see 4.2.696), a `Symmetry_tolerance` (see 4.2.883), or a `Total_runout_tolerance` (see 4.2.929).

The data associated with a `Geometric_tolerance_with_referenced_datum_system` are the following:

— `datum_system_definition`.

4.2.422.1 datum_system_definition

The `datum_system_definition` specifies the `Datum_system_defining_conditions` (see 4.2.237) that define the `Datum_system` (see 4.2.236) that in relation to which the `Geometric_tolerance_with_referenced_datum_system` specifies or partially specifies design requirements.

Constraints:

WR1: The `Geometric_tolerance_with_referenced_datum_system` shall be specified by at most one `Separate_geometric_tolerance_requirement` as one of the `separated_geometric_tolerances`.

NOTE EXPRESS specification:

```

*)
ENTITY geometric_tolerance_with_referenced_datum_system
  ABSTRACT SUPERTYPE OF (ONEOF (angularity_tolerance,
    parallelism_tolerance, perpendicularity_tolerance,
    circular_runout_tolerance, symmetry_tolerance,
    concentricity_and_coaxiality_tolerance,
    profile_of_any_line_tolerance_with_referenced_datum_system,
    profile_of_any_surface_tolerance_with_referenced_datum_system,
    position_tolerance_with_referenced_datum_system,
    total_runout_tolerance))
  SUBTYPE OF (geometric_tolerance);
    datum_system_definition : datum_system_defining_conditions;
END_ENTITY; -- geometric_tolerance_with_referenced_datum_system

( *

```

4.2.423 Geometric_tolerance_without_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Geometric_tolerance_without_referenced_datum_system` is a type of `Geometric_tolerance` (see 4.2.419) that specifies or partially specifies the design requirement on one or more `Datum_system` (see 4.2.236) independent geometric characteristics such as flatness of the `Shape_element` (see 4.2.810) to which the `Geometric_tolerance_without_referenced_datum_system` is applied. A `Geometric_tolerance_without_referenced_datum_system` is one of a `Circularity_tolerance` (see 4.2.103), a `Cylindricity_tolerance` (see 4.2.210), a `Flatness_tolerance` (see 4.2.386), a `Position_tolerance_without_referenced_datum_system` (see 4.2.697), a `Profile_of_any_line_tolerance_without_referenced_datum_system` (see 4.2.746), a `Profile_of_any_surface_tolerance_without_referenced_datum_system` (see 4.2.749), or a `Straightness_tolerance` (see 4.2.844).

NOTE EXPRESS specification:

```

*)
ENTITY geometric_tolerance_without_referenced_datum_system
  ABSTRACT SUPERTYPE OF (ONEOF (circularity_tolerance,
    cylindricity_tolerance, flatness_tolerance, straightness_tolerance,
    position_tolerance_without_referenced_datum_system,
    profile_of_any_surface_tolerance_without_referenced_datum_system,
    profile_of_any_line_tolerance_without_referenced_datum_system))
  SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- geometric_tolerance_without_referenced_datum_system

--tt tc1 wd9 modified
( *

```

4.2.424 Geometrically_bounded_2d_wireframe

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Geometrically_bounded_2d_wireframe is a type of Feature_shape_definition (see 4.2.372). A Geometrically_bounded_2d_wireframe is a geometrically bounded wireframe with a dimensionality equal to 2.

NOTE The Geometrically_bounded_2d_wireframe, being a wireframe-based feature, does not support the representation of areas.

NOTE EXPRESS specification:

```
*)
ENTITY geometrically_bounded_2d_wireframe
  SUBTYPE OF (planar_shape);
END_ENTITY; -- geometrically_bounded_2d_wireframe

( *
```

4.2.425 Geometrically_bounded_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Geometrically_bounded_surface is a type of Bound_volume_shape (see 4.2.76). The Geometrically_bounded_surface is a surface that does not include topological information as to what edges are shared between surfaces. Only geometric information is supplied, which is useful only for presentation purposes. This entity should not be used to convey connectivity-related data unless independent connectivity reconstruction and verification is performed.

NOTE EXPRESS specification:

```
*)
ENTITY geometrically_bounded_surface
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- geometrically_bounded_surface

( *
```

4.2.426 Guided_wave_bare_die_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Guided_wave_bare_die_terminal is a type of Minimally_defined_bare_die_terminal (see 4.2.545) used with guided waves in high frequency or fiber-optic applications.

NOTE EXPRESS specification:

```
*)
ENTITY guided_wave_bare_die_terminal
  SUBTYPE OF (minimally_defined_bare_die_terminal);
END_ENTITY; -- guided_wave_bare_die_terminal

( *
```

4.2.427 Guided_wave_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Guided_wave_terminal is a type of Package_terminal (see 4.2.581) that is explicitly included for the purpose of providing access to the functionality of a Packaged_part (see 4.2.591) using guided waves.

EXAMPLE 1 Examples of Guided_wave_terminal include fiber-optic terminals and waveguide ports.

Constraints:

WR1: A Guided_wave_terminal shall not be a Wire_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY guided_wave_terminal
  SUBTYPE OF (package_terminal);
END_ENTITY; -- guided_wave_terminal

( * tt tc1 wd9 deleted
--tt tc1 correction
( *
```

4.2.428 Illuminance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Illuminance_data_element is a

The data associated with an Illuminance_data_element are the following:

— additional_data.

4.2.428.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY illuminance_data_element  
  SUBTYPE OF (data_element);  
END_ENTITY; -- illuminance_data_element  
  
(*
```

4.2.429 Inductance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Inductance_data_element is a

The data associated with an Inductance_data_element are the following:

— additional_data.

4.2.429.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY inductance_data_element  
  SUBTYPE OF (data_element);  
END_ENTITY; -- inductance_data_element  
  
(*
```

4.2.430 Inspection_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inspection_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that specifies a production inspection related requirement. An Inspection_requirement_occurrence may be a Layout_line_width_tolerance_requirement_occurrence (see 4.2.503) or a Layout_land_width_tolerance_requirement_occurrence (see 4.2.502) or a Fabrication_thickness_requirement_occurrence (see 4.2.370).

NOTE EXPRESS specification:

```
*)
ENTITY inspection_requirement_occurrence
  SUPERTYPE OF(ONEOF(layout_line_width_tolerance_requirement_occurrence,
    layout_land_width_tolerance_requirement_occurrence,
    fabrication_thickness_requirement_occurrence))
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- inspection_requirement_occurrence
(*
```

4.2.431 Inter_stratum_extent

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inter_stratum_extent is a type of Managed_design_object (see 4.2.524). An Inter_stratum_extent is the extent of Stratum (see 4.2.845) specified for a design purpose. The extent includes the from_stratum and to_stratum.

The data associated with an Inter_stratum_extent are the following:

- from_stratum;
- to_stratum.

4.2.431.1 from_stratum

The from_stratum specifies the Stratum (see 4.2.845) at one end of the Inter_stratum_extent. This Stratum is the Stratum closer to the Design_layer_stratum (see 4.2.275) whose primary_design_layer_stratum attribute value is TRUE than is the to_stratum.

NOTE The from_stratum by convention is closest to the primary side of the assembly and furthest from the secondary side of the assembly.

4.2.431.2 to_stratum

The to_stratum specifies the Stratum (see 4.2.845) at one end of the Inter_stratum_extent. This Stratum is further from the Design_layer_stratum (see 4.2.275) whose primary_design_layer_stratum attribute value is TRUE than is the from_stratum.

NOTE The to_stratum by convention is furthest from the primary side of the assembly, and closest to the secondary side of the assembly.

Constraints:

WR1: The members of stratum specified by from_stratum and to_stratum shall be associated with the same definition of the same product.

NOTE When the from_stratum and to_stratum are the same Stratum the extent of the Inter_stratum_extent is that Stratum.

NOTE EXPRESS specification:

```
*)
ENTITY inter_stratum_extent
  SUBTYPE OF (managed_design_object);
    from_stratum : stratum;
    to_stratum   : stratum;
END_ENTITY; -- inter_stratum_extent

( *
```

4.2.432 Inter_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inter_stratum_feature is a type of Laminate_component (see 4.2.483). An Inter_stratum_feature is one of a Cutout (see 4.2.208), a Cutout_edge_segment (see 4.2.209), a Dielectric_material_passage (see 4.2.289), an Interconnect_module_edge (see 4.2.449), an Interconnect_module_edge_segment (see 4.2.450), a Plated_inter_stratum_feature (see 4.2.687), or an Unsupported_passage (see 4.2.939). An Inter_stratum_feature spans one or more Stratum (see 4.2.845).

NOTE 1 An Inter_stratum_feature may be realized by removal of material by drilling, routing, punching, chemically etching, laser burning, or otherwise penetrating one or more sequential Stratum; by the addition of material; or by combining both removal and addition of material in the fabrication process.

NOTE 2 Keep out or keep in regions can be specified by populating Interconnect_module_constraint_region and its optional set of Interconnect_module_design_object_select. The Interconnect_module_design_object_select can select an Inter_startum_feature as the design feature of interest.

The data associated with an `Inter_stratum_feature` are the following:

- `feature_of_size`;
- `usage_definition`.

NOTE Minimum and maximum material conditions are related to `Inter_stratum_feature` by its attribute `usage_definition`. The maximum and minimum material conditions established keep-out areas. Since these maximum and minimum conditions reflect a fabrication process, the designer does not have this area to locate other features. A situation such as tight spacing of conductors on a PWB must take into account the maximum and minimum material condition or the design will not be optimally realized through the fabrication process.

4.2.432.1 `feature_of_size`

The `feature_of_size` specifies whether or not the `Inter_stratum_feature` interfaces with a feature of another product in an assembly context. A value of `TRUE` specifies that it interfaces. A value of `FALSE` specifies that it does not interface.

NOTE Certain tolerance information depends on knowledge of a particular feature interfacing with features of a mating product. The maximum material conditions of the `Inter_stratum_feature` and of the feature of the mating product are taken into consideration for calculating the fit at assembly.

4.2.432.2 `usage_definition`

The `usage_definition` specifies the role of the `Inter_stratum_feature_template` (see 4.2.435) for the `Inter_stratum_feature`.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_feature
  ABSTRACT SUPERTYPE OF (ONEOF (unsupported_passage,
    plated_inter_stratum_feature, cutout, interconnect_module_edge,
    interconnect_module_edge_segment, cutout_edge_segment,
    dielectric_material_passage))
  SUBTYPE OF (laminate_component);
    feature_of_size : BOOLEAN;
    SELF\assembly_component.usage_definition : inter_stratum_feature_template;
END_ENTITY; -- inter_stratum_feature
( *
```

4.2.433 `Inter_stratum_feature_edge_segment_template`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Inter_stratum_feature_edge_segment_template` is a type of `Inter_stratum_feature_template` (see 4.2.435).

The data associated with an `Inter_stratum_feature_edge_segment_template` are the following:

- `composed_edge`;
- `end_vertex`;
- `start_vertex`.

4.2.433.1 composed_edge

The `composed_edge` specifies the role of the `Inter_stratum_feature_edge_template` (see 4.2.434) the `Inter_stratum_feature_edge_segment_template` is helping to compose.

4.2.433.2 end_vertex

The `end_vertex` specifies one role of the `Edge_segment_vertex` (see 4.2.312) for the `Inter_stratum_feature_edge_segment_template`.

4.2.433.3 start_vertex

The `start_vertex` specifies one role of the `Edge_segment_vertex` (see 4.2.312) for the `Inter_stratum_feature_edge_segment_template`.

Constraints:

WR1: The `end_vertex` shall not be the `start_vertex`.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_feature_edge_segment_template
  SUBTYPE OF (inter_stratum_feature_template);
    composed_edge : inter_stratum_feature_edge_template;
    start_vertex   : edge_segment_vertex;
    end_vertex     : edge_segment_vertex;
    reference_path : trimmed_line;
    cross_section  : OPTIONAL edge_segment_cross_section;
WHERE
  WR1: EXISTS (cross_section) XOR (SIZEOF (
    USEDIN(SELF\part_template,
      'AP210_ARM.PART_TEMPLATE_3D_SHAPE.' +
      'SHAPE_CHARACTERIZED_PART_TEMPLATE')
  ) &#62;= 0);
  (* start_vertex and end_vertex associated points shall be in the domain of the path. *)
  WR2: EXISTS(reference_path.start_point) AND EXISTS(reference_path.end_point);
  WR3: NOT(EXISTS(start_vertex\datum_point.location_in_2d)) OR
    (reference_path.start_point == start_vertex\datum_point.location_in_2d);
  WR4: NOT(EXISTS(start_vertex\datum_point.location_in_3d)) OR
    (reference_path.start_point == start_vertex\datum_point.location_in_3d);
  WR5: NOT(EXISTS(end_vertex\datum_point.location_in_2d)) OR
    (reference_path.end_point == start_vertex\datum_point.location_in_2d);
  WR6: NOT(EXISTS(end_vertex\datum_point.location_in_3d)) OR
    (reference_path.end_point == start_vertex\datum_point.location_in_3d);
  WR7: start_vertex == end_vertex;
END_ENTITY; -- inter_stratum_feature_edge_segment_template

( *

```

4.2.434 Inter_stratum_feature_edge_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inter_stratum_feature_edge_template is a type of Inter_stratum_feature_template (see 4.2.435).

The data associated with an Inter_stratum_feature_edge_template are the following:

— edges.

4.2.434.1 edges

The edges is an inverse relationship that specifies that the existence of the Inter_stratum_feature_edge_template is dependent on the existence of the Inter_stratum_feature_edge_segment_template (see 4.2.433) that specifies the Inter_stratum_feature_edge_template as its composed_edge. There shall be two or more Inter_stratum_feature_edge_segment_template for an Inter_stratum_feature_edge_template.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_feature_edge_template
  SUBTYPE OF (inter_stratum_feature_template);
INVERSE
  edges : SET [2:?] OF inter_stratum_feature_edge_segment_template for composed_edge;
END_ENTITY; -- inter_stratum_feature_edge_template

( *

```

4.2.435 Inter_stratum_feature_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inter_stratum_feature_template is a type of Part_template (see 4.2.620). An Inter_stratum_feature_template may be one of a Component_termination_passage_template (see 4.2.150), an Unsupported_passage_template (see 4.2.941), or a Via_template (see 4.2.952). An Inter_stratum_feature_template is a definitional construct that defines the cross-section shape of layered constructs.

The data associated with an Inter_stratum_feature_template are the following:

— of_passage_technology.

4.2.435.1 of_passage_technology

The of_passage_technology specifies the role of the Passage_technology (see 4.2.638) for the Inter_stratum_feature_template. The op_passage_technology specifies the material and layered properties for the Inter_stratum_feature_template.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_feature_template
  SUPERTYPE OF (ONEOF (component_termination_passage_template,
    via_template, unsupported_passage_template,
    inter_stratum_feature_edge_segment_template,
    inter_stratum_feature_edge_template))
  SUBTYPE OF (part_template);
  of_passage_technology : passage_technology;
END_ENTITY; -- inter_stratum_feature_template

( *

```

4.2.436 Inter_stratum_join_implementation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Inter_stratum_join_implementation` is a type of `Managed_design_object` (see 4.2.524). An `Inter_stratum_join_implementation` is one of a `Physical_network_supporting_inter_stratum_feature` (see 4.2.661) or a `Physical_network_supporting_stratum_feature_conductive_join` (see 4.2.662). An `Inter_stratum_join_implementation` is a physical feature helping to implement a `Physical_network` (see 4.2.660). An `Inter_stratum_join_implementation` is not intended to be implemented alone. It is one of several Application Objects that together form the physical network implementation model.

NOTE 1 The types of `Inter_stratum_join_implementation` correspond to two physical implementation mechanisms.

NOTE 2 The first mechanism consists of creating a passage or edge in the material and then adding conductive material to the passage or edge wall and features on the `to_stratum` and `from_stratum` of the `Inter_stratum_extent` (see 4.2.431). This mechanism results in a connection between the required features implemented with the passage or edge wall.

NOTE 3 The second mechanism does not result in a conducting passage, but relies on a bond (achieved either by direct deposition or some other means) between the features related to the specified `layer_connection_points`. The second mechanism usually relies on a passage in only the dielectric material separating the bonded materials. A cross-sectional view of the pcb may show a vertical displacement of strata in the area of this object, due to the fact that the dielectric Stratum (see 4.2.845) is much thicker than the conductive Stratum. The shape of the vertical displacement is not a part of the definition of the `Inter_stratum_join_implementation` when the second mechanism is used.

The data associated with an `Inter_stratum_join_implementation` are the following:

— `topological_requirement`.

4.2.436.1 topological_requirement

The `topological_requirement` specifies the role of the `Inter_stratum_join_relationship` (see 4.2.437) for the `Inter_stratum_join_implementation`.

NOTE The following is a description of a realization of an `Inter_stratum_join_implementation` appropriate for some technologies. Features that implement inter layer connection are realized by forming passages in the sequence of stratum that define the extent of the passage. The passage has a nominally uniform cross-section and has a direction normal to the surface specified as the `from_stratum`. If that surface is not planar, then the surface coordinates of the layer connection points vary layer to layer to account for the finite thickness of the stratum comprising the Stratum.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_join_implementation
  ABSTRACT SUPERTYPE OF (ONEOF (
    physical_network_supporting_stratum_feature_conductive_join,
    physical_network_supporting_inter_stratum_feature))
  SUBTYPE OF (managed_design_object);
  topological_requirement : inter_stratum_join_relationship;
END_ENTITY; -- inter_stratum_join_implementation

--tt tc1 wd8 modified
( *

```

4.2.437 Inter_stratum_join_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Inter_stratum_join_relationship is a type of Join_relationship (see 4.2.479), in which the elements to be joined, Layer_connection_point (see 4.2.499), shall be on different Design_layer_stratum (see 4.2.275). An Inter_stratum_join_relationship is a requirement for a Inter_stratum_join_implementation (see 4.2.436).

NOTE In order for a design to be complete the Inter_stratum_join_implementation that specifies Inter_stratum_join_relationship in the role of topological_requirement must be populated.

Constraints:

WR1: No member of Design_layer_stratum shall be the resident_stratum for more than one member of Layer_connection_point that is joined by the same member of the Inter_stratum_join_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY inter_stratum_join_relationship
  SUBTYPE OF (join_relationship);
  points_to_be_connected : SET [2:?] OF layer_connection_point;
END_ENTITY; -- inter_stratum_join_relationship

( *

```

4.2.438 Interconnect_component_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_component_interface_terminal` is a type of `Component_interface_terminal` (see 4.2.125) that is some aspect of a generic `Assembly_component` (see 4.2.34) participate in a connection for a `Pca` (see 4.2.639). The `Interconnect_component_interface_terminal` is defined in-situ. There is no `Part_feature` (see 4.2.609) associated with it.

EXAMPLE 1 An example of an `Interconnect_component_interface_terminal` is a mechanical conductive fastener that attaches a flexible wire lead lug to the pcb. Current flows through the fastener because it is part of the conduction path for a signal.

EXAMPLE 2 A plain wire is considered to be an interconnect component as well. This `Interconnect_component_interface_terminal` is defined in-situ. There is no `Part_feature` (see 4.2.609) associated with it. This is the opposite end of the wire from that defined for the same wire in `Interconnect_component_join_terminal` (see 4.2.439).

The data associated with an `Interconnect_component_interface_terminal` are the following:

- `connection`;
- `name`.

4.2.438.1 connection

The `connection` specifies the role of the `Connection_zone` (see 4.2.185) for the `Interconnect_component_interface_terminal`. The `connection` is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the data shall be provided.

4.2.438.2 name

The `name` specifies the string for the human-interpretable identifier for this `Interconnect_component_interface_terminal`.

Constraints:

UR1: The combination of the `name` and the inherited `associated_component` shall be unique within a population of `Interconnect_component_interface_terminal`.

WR1: The shapes that describe the `connection_zone` specified by `connections` shall reside on one or more surfaces of the `assembly_component` in the role of `associated_component` specified by the super-type `component_feature`.

WR2: An `Interconnect_component_interface_terminal` shall participate in only one `Assembly_joint` (see 4.2.41).

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_component_interface_terminal
  SUBTYPE OF (component_interface_terminal);
  connection : OPTIONAL connection_zone;
UNIQUE
  UR1 : SELF\shape_element.name, SELF\component_feature.associated_component;
END_ENTITY; -- interconnect_component_interface_terminal

( *

```

4.2.439 Interconnect_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_component_join_terminal is a type of Component_terminal (see 4.2.137) that is some aspect of a generic Assembly_component (see 4.2.34) participating in connectivity for a Pca (see 4.2.639). This Interconnect_component_join_terminal is defined in-situ.

EXAMPLE 1 An example of an Interconnect_component_join_terminal is a mechanical conductive fastener that attaches a flexible wire lead lug to the pcb. Current flows through the fastener because it is part of the conduction path for a signal.

EXAMPLE 2 A plain wire is considered to be a mechanical component as well. There is no Part_feature (see 4.2.609) associated with it. This is the opposite end of the wire from that defined for the same wire in Interconnect_component_interface_terminal (see 4.2.438).

The data associated with an Interconnect_component_join_terminal are the following:

- connection;
- name.

4.2.439.1 connection

The connection specifies the role of the Connection_zone (see 4.2.185) for the Interconnect_component_join_terminal. The connection is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the data shall be provided.

4.2.439.2 name

The name specifies the string for the human-interpretable identifier for this Interconnect_component_join_terminal.

Constraints:

UR1: The combination of the name and the inherited associated_component shall be unique within a population of Interconnect_component_join_terminal.

WR1: The shapes that describe the connection_zone specified by connections shall reside on one or more surfaces of the assembly_component in the role of associated_component specified by the super-type Component_feature of this terminal.

WR2: Interconnect_component_join_terminal shall participate in no more than one Assembly_joint.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_component_join_terminal
  SUBTYPE OF (physical_component_terminal);
  connection : OPTIONAL connection_zone;
  SELF\component_feature.associated_component : physical_component;
  UNIQUE
    UR1 : SELF\shape_element.name,
          SELF\component_feature.associated_component;
END_ENTITY; -- interconnect_component_join_terminal
( *

```

4.2.440 Interconnect_module

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module is a type of Physical_unit_design_view (see 4.2.667). An Interconnect_module may be a Pcb (see 4.2.642). An Interconnect_module is a design view of a product. An Interconnect_module may be an aggregation of Stratum (see 4.2.845). An Interconnect_module may be an aggregation of Interconnect_module_component (see 4.2.442) An Interconnect_module features are defined by Stratum_feature (see 4.2.851), within the bounds established by Inter_stratum_feature (see 4.2.432).

NOTE Not all products defined by this design view are classified in general as interconnect substrates.

EXAMPLE 1 A package may be defined by the design owners using the Interconnect_module design view.

EXAMPLE 2 A ribbon cable, minus the connectors, may be defined by the design owners using the Interconnect_module design view. Complex 3d arrangements of the ribbon may be supported.

EXAMPLE 3 Figure 49 illustrates two categories of complex land patterns. For a specific technology implementation in an Interconnect_module, all parameters would be assigned specific values, in addition to the explicit geometry and material descriptions otherwise provided.

EXAMPLE 4 Figure 50 illustrates two categories of simple land patterns. For a specific technology implementation in an Interconnect_module, all parameters would be assigned specific values, in addition to the explicit geometry and material descriptions otherwise provided.

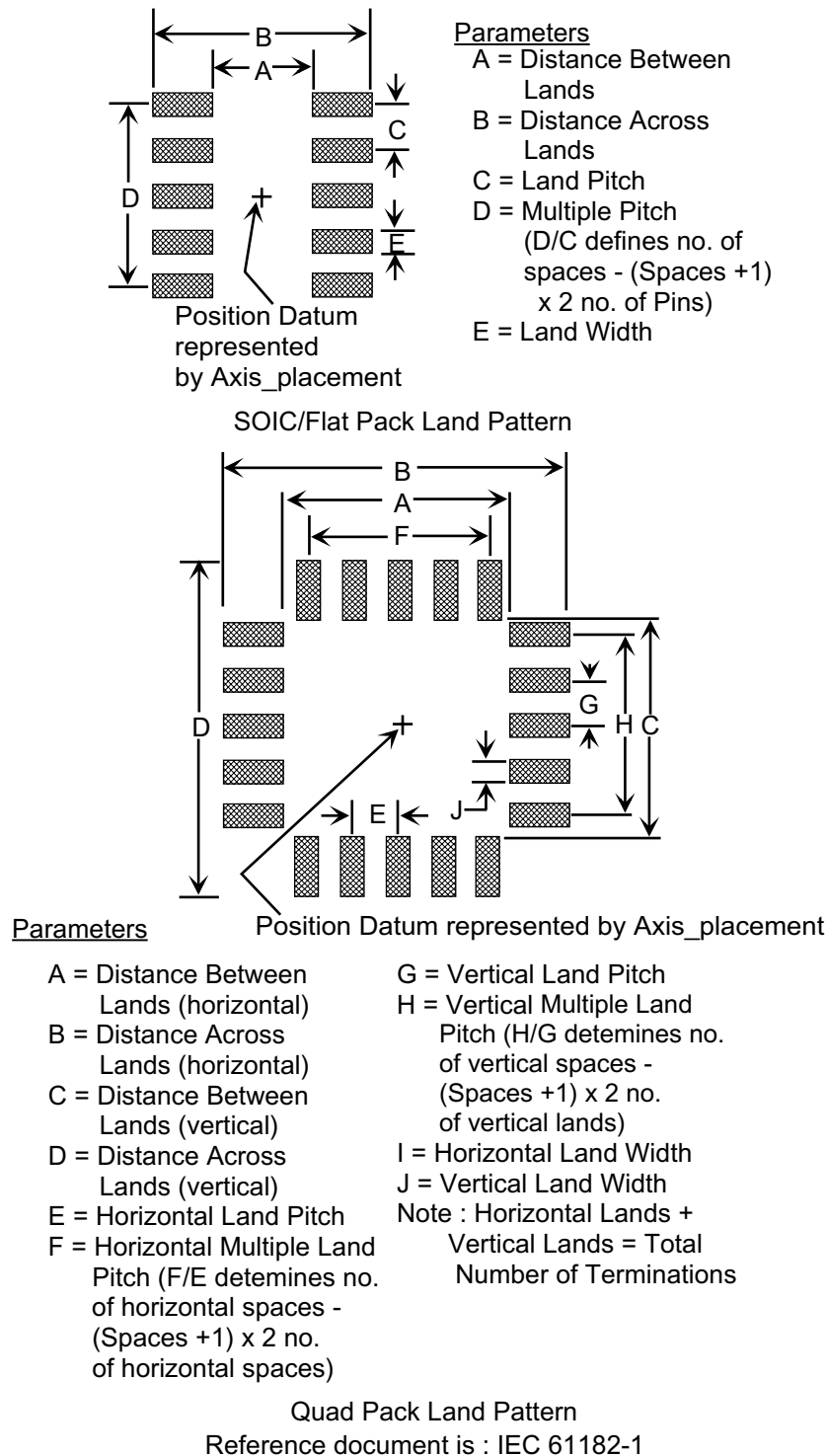
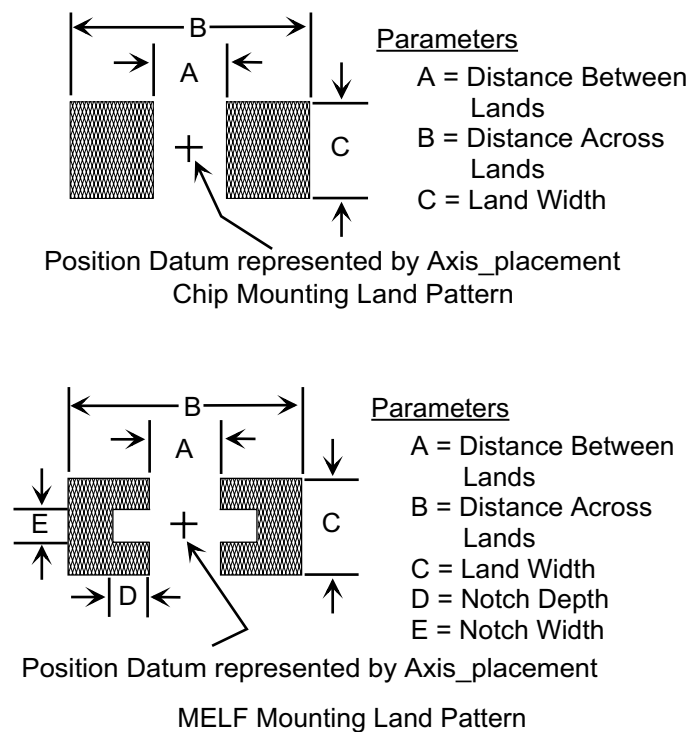


Figure 49 – Complex technology specific land pattern definition



Reference document is : IEC 61182-1

Figure 50 – Simple technology specific land pattern definition

The data associated with an Interconnect_module are the following:

— assembly_design_requirement.

4.2.440.1 assembly_design_requirement

The assembly_design_requirement specifies the Assembly_module (see 4.2.43) for the Interconnect_module as a specified source of requirements not otherwise explicitly allocated to the Interconnect_module. This requirement shall be interpreted as establishing a direct relationship to the assembly or assemblies that define data that may be used as requirements for the Interconnect_module. There may be more than one Assembly_module for a Interconnect_module. The assembly_design_requirement is optional and need not be supplied.

NOTE This relationship is provided so that design requirements may be traced to the original source even though an Interconnect_module may be designed into many subsequent assembly designs.

NOTE Explicit data requirements may be contained in one or more assembly design views. These multiple requirements can be specified by the the `Component_external_reference` and `Component_feature_external_reference` application objects. Component features which are contained in one design view may be associated to layout features that are contained in another design view. `Component_external_reference` and `Component_feature_external_reference` maintain this association while also maintaining the completeness of the design view.

EXAMPLE 1 If the attribute name specified by `Ee_product_category` (see 4.2.320) that is in the role of `product_classification` specified by `Ee_product` (see 4.2.319) that is in the role of `versioned_item` specified by `Ee_product_version` (see 4.2.328) that is in the role of `version` specified by the supertype `Ee_product_definition` is "Interconnect module", and if the sub category of the `Ee_product_category` is "land pattern", then the purpose for inclusion is to provide land pattern source data that may be manufacturer specific or industry standard. The attribute `standard_product_indicator` specified by the same member of `Ee_product` shall be true for industry standard pattern and material designation. The material designations, arrangement, etc., are included as part of this definition since this application object is a design view. The instantiation of the land pattern into the design is accomplished using the `Component_make_from_relationship` (see 4.2.126) to specify each component of the land pattern as it is reused in the design. The Application Objects `Padstack_definition` (see 4.2.595) and `Footprint_definition` (see 4.2.388) are provided to allow specific usages of patterns generally understood in industry to be exchanged. It is suggested to use a `Footprint_definition` to exchange land patterns instead of the generic mechanism.

Constraints:

WR1: `Interconnect_module.usage_view` (this is an inherited attribute from `Physical_unit_design_view`) shall be either an `Interconnect_module_usage_view` or a `Package`.

WR2: There shall be exactly one primary design layer stratum in the `Interconnect_module`.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module
  SUPERTYPE OF (ONEOF(
    pcb,
    reference_packaged_part_interconnect_implementation))
  SUBTYPE OF (physical_unit_design_view);
  assembly_design_requirement : OPTIONAL SET [1:?] OF assembly_module;
INVERSE
  of_fabrication_thickness_requirement : SET [0:?] OF requirement_allocation
    FOR target_product_object;
WHERE
  WR1: SIZEOF(QUERY( ftr &#60;* of_fabrication_thickness_requirement |
    NOT('AP210_ARM.FABRICATION_THICKNESS_REQUIREMENT_OCCURRENCE' IN
    TYPEOF(ftr\requirement_allocation.allocated_requirement))
    )) = 0;
  WR2: SIZEOF(QUERY(imsar &#60;* USEDIN(SELF,
    'AP210_ARM.INTERCONNECT_MODULE_STRATUM_ASSEMBLY_RELATIONSHIP.' +
    'ASSEMBLY') |
    (('AP210_ARM.DESIGN_LAYER_STRATUM' IN
    TYPEOF(imsar.component)) AND
    (imsar.component\design_layer_stratum.primary_design_layer_stratum = TRUE))
    )) = 1;
END_ENTITY; -- interconnect_module
( *

```

4.2.441 Interconnect_module_cavity_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Interconnect_module_cavity_surface is a type of Interconnect_module_surface_feature (see 4.2.462).

The data associated with an Interconnect_module_cavity_surface are the following:

— reference_surface.

4.2.441.1 reference_surface

The reference_surface specifies an Interconnect_module_surface_feature (see 4.2.462) for the Interconnect_module_cavity_surface.

Constraints:

WR1: The reference_surface shall not be this Interconnect_module_cavity_surface.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_cavity_surface
  SUBTYPE OF (interconnect_module_surface_feature);
  reference_surface : interconnect_module_surface_feature;
WHERE
  WR1: SELF :&#60;&#62;; reference_surface;
END_ENTITY; -- interconnect_module_cavity_surface

( *

```

4.2.442 Interconnect_module_component

An `Interconnect_module_component` is a type of `Assembly_component` (see 4.2.34). An `Interconnect_module_component` is an instance of an `Interconnect_module_usage_view`.

NOTE The `Interconnect_module_component` can be, for example, a circuit board in an assembly. Alternatively, it could be the substrate for a multi-chip module. This entity is included specifically so that the assembly design definition can be queried directly to determine which component in the assembly is the substrate. The `Interconnect_module_component` is likely the target of connectivity requirement information as allocated to various components. An `Interconnect_module_component` does not include all the internal details of the design of the `Interconnect_module` (see 4.2.440).

The data associated with an `Interconnect_module_component` are the following:

— `usage_definition`.

4.2.442.1 usage_definition

The `usage_definition` specifies the role of the `Interconnect_module_usage_view` (see 4.2.465) for the `Interconnect_module_component`.

NOTE The `usage_view` specifies the definitions of the terminals and other interface elements of the `Interconnect_module` (see 4.2.440).

Constraints:

WR1: An `Interconnect_module_component` shall not be mounted onto itself.

NOTE EXPRESS specification:

```
*)
ENTITY interconnect_module_component
  SUPERTYPE OF(interconnect_module_macro_component)
  SUBTYPE OF (physical_component);
  SELF\assembly_component.usage_definition : interconnect_module_usage_view;
END_ENTITY; -- interconnect_module_component
(*
```

4.2.443 Interconnect_module_component_stratum_based_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_component_stratum_based_terminal is a type of Interconnect_module_component_terminal (see 4.2.445) included to support the case where explicit Interconnect_module_component_terminals are being connected by an Interconnect_module_product_assembly_view_fabrication_joint (see 4.2.458) as part of a sequential laminate assembly view for design or analysis purposes.

NOTE A sequential laminate assembly description may be necessary to support embedded discrete components.

The data associated with an Interconnect_module_component_stratum_based_terminal are the following:

— definition.

4.2.443.1 definition

The definition specifies the role of the Interconnect_module_stratum_based_terminal (see 4.2.461) for the Interconnect_module_component_stratum_based_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY interconnect_module_component_stratum_based_terminal
  SUBTYPE OF (interconnect_module_component_terminal);
  SELF\component_feature.definition : interconnect_module_stratum_based_terminal;
END_ENTITY; -- interconnect_module_component_stratum_based_terminal
(*
```

4.2.444 Interconnect_module_component_surface_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_component_surface_feature` is a type of `Component_feature` (see 4.2.118). An `Interconnect_module_component_surface_feature` is a surface or a segment of the edge of the `Interconnect_module_component` (see 4.2.442) identified for assembly purposes. The definition attribute inherited from `Component_feature` (see 4.2.118) shall specify a type of `Interconnect_module_surface_feature` (see 4.2.462). The associated_component attribute inherited from `Component_feature` shall specify a member of `Interconnect_module_component` (see 4.2.442). An `Interconnect_module_component_surface_feature` may also be an `Interconnect_module_component_terminal` (see 4.2.445), but shall not be any other type of `Component_feature`.

NOTE The entire primary or secondary surface of the `Interconnect_module_component` may be considered to be a feature in this usage.

NOTE EXPRESS specification:

```
*)
ENTITY interconnect_module_component_surface_feature
  SUBTYPE OF (physical_component_feature);
--tt tc1 wd11 modify
  SELF\component_feature.definition : interconnect_module_surface_feature;
END_ENTITY; -- interconnect_module_component_surface_feature

--tt tc1 wd9 modify
(*
```

4.2.445 `Interconnect_module_component_terminal`

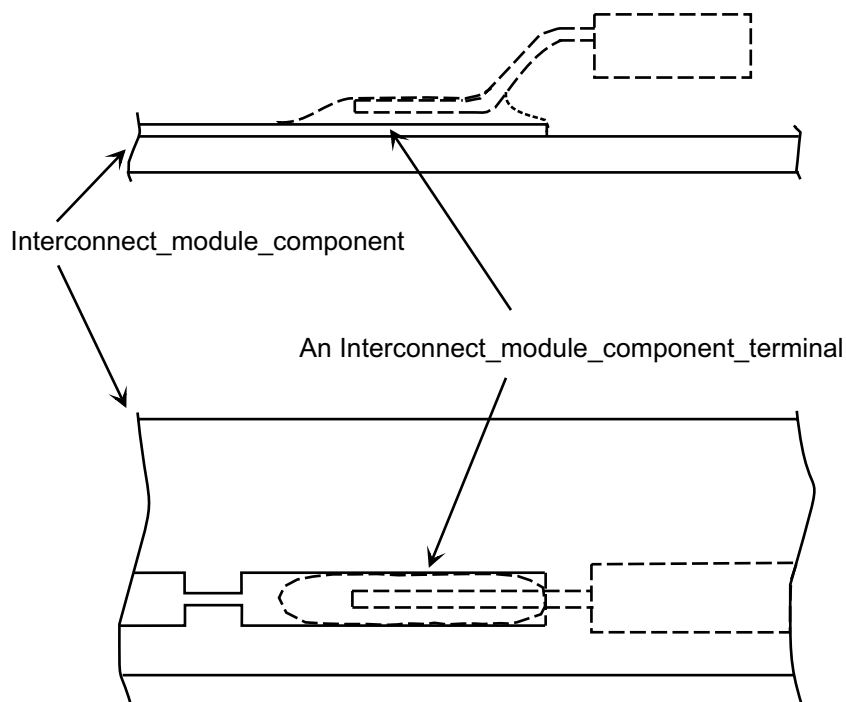
NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_component_terminal` is a type of `Component_terminal` (see 4.2.137). An `Interconnect_module_component_terminal` may be an `Interconnect_module_component_stratum_based_terminal` (see 4.2.443). A member of `Interconnect_module_component_terminal` may be referenced by a member of `Assembly_joint` to describe the partial or complete implementation of a connection defined as a requirement by a member of `Physical_connectivity_definition`.

EXAMPLE 1 Figure 51 illustrates the relationship between an `Interconnect_module_component_terminal` and its mating feature in an assembly.

The data associated with an `Interconnect_module_component_terminal` are the following:

- associated_component;
- definition.



Note : The design view associated with this data is the Assembly_module

Figure 51 – Interconnect_module_component_terminal in an assembly

4.2.445.1 associated_component

The associated_component specifies the role of the Interconnect_module_component (see 4.2.442) for the Interconnect_module_component_terminal.

4.2.445.2 definition

The definition specifies the role of the Interconnect_module_terminal (see 4.2.463) for the Interconnect_module_component_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_component_terminal
  SUPERTYPE OF ((interconnect_module_component_stratum_based_terminal)
                ANDOR (interconnect_module_macro_component_join_terminal))
  SUBTYPE OF (physical_component_terminal);
  SELF\component_feature.definition          : interconnect_module_terminal;
  SELF\component_feature.associated_component : interconnect_module_component;
END_ENTITY; -- interconnect_module_component_terminal

( *

```

4.2.446 Interconnect_module_constraint_region

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_constraint_region is a type of Non_feature_shape_element (see 4.2.560). An Interconnect_module_constraint_region is a region within an Interconnect_module (see 4.2.440) in which a placement constraint on some category of design object is to be met by the design. An Interconnect_module_constraint_region may overlap other Interconnect_module_constraint_regions.

The data associated with an Interconnect_module_constraint_region are the following:

- associated_stratum_extent;
- constrained_design_object_category;
- design_specific_purpose;
- keepout;
- non_conformant_interconnect_module_design_object;
- precedent_version_constraint_region;
- requirement;
- scope.

4.2.446.1 associated_stratum_extent

The associated_stratum_extent specifies the Inter_stratum_extent (see 4.2.431) as the extent of Stratum (see 4.2.845) for the Interconnect_module_constraint_region. There shall be one or more Inter_stratum_extent for an Interconnect_module_constraint_region. The Inter_stratum_extent shall all be for the same scope.

4.2.446.2 **constrained_design_object_category**

The `constrained_design_object_category` specifies either an `interconnect_module_cutout`, an `interconnect_module_fill_area`, an `interconnect_module_inter_stratum_feature`, an `interconnect_module_stratum_feature`, or an `interconnect_module_via` as a category for the `Interconnect_module_constraint_region`. There may be more than one `interconnect_module_cutout`, `interconnect_module_fill_area`, `interconnect_module_inter_stratum_feature`, or `interconnect_module_via` for an `Interconnect_module_constraint_region`. The names shall be interpreted as the identification of Application Object categories to be constrained.

4.2.446.3 **design_specific_purpose**

The `design_specific_purpose` identifies the purpose the enterprise defined for the `Interconnect_module_constraint_region`.

4.2.446.4 **keepout**

The `keepout` specifies if the meaning of the `Interconnect_module_constraint_region` is to describe that items shall be in the interior of the `element_shape` or shall be on the exterior of the `element_shape`. The value of `TRUE` states that objects in the `constrained_design_object_category` are not to be within the `element_shape`. The value of `FALSE` states that objects in the `constrained_design_object_category` are to be only within the `element_shape`.

4.2.446.5 **non_conformant_interconnect_module_design_object**

The `non_conformant_interconnect_module_design_object` specifies either a `Component_termination_passage` (see 4.2.143), a `Conductive_filled_area` (see 4.2.170), a `Conductor` (see 4.2.176), a `Cutout` (see 4.2.208), an `Inter_stratum_feature` (see 4.2.432), a `Laminate_component` (see 4.2.483), a `Land` (see 4.2.490), a `Non_functional_land` (see 4.2.562), a `Stratum_feature` (see 4.2.851), or a `Via` (see 4.2.947) for the `Interconnect_module_constraint_region`. There may be more than one `Component_termination_passage`, `Conductive_filled_area`, `Conductor`, `Cutout`, `Inter_stratum_feature`, `Laminate_component`, `Land`, `Non_functional_land`, `Stratum_feature` (see 4.2.851), or `Via` for an `Interconnect_module_constraint_region`.

4.2.446.6 **precedent_version_constraint_region**

The `precedent_version_constraint_region` is an inverse relationship that specifies that the `Interconnect_module_constraint_region` is existence dependent on a specific `Interconnect_module_constraint_region` as a precedent region. The usage of this relationship is to enable traceability to original `Interconnect_module_constraint_region` during design iterations.

NOTE Since the `Interconnect_module_constraint_region` is dependent on a specific `Interconnect_module`, it is not possible to use the mechanism of associating one `Interconnect_module_constraint_region` with several `Interconnect_module`, each of which is for a different design version. In addition, the `non_conformant_interconnect_module_design_object` is for objects within the scope, which also precludes direct re-use of the same `Interconnect_module_constraint_region`. It is expected that only the first `Interconnect_module_constraint_region` populated would have an associated shape to eliminate the need for carrying redundant shape data.

4.2.446.7 requirement

The requirement specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Interconnect_module_constraint_region.

4.2.446.8 scope

The scope specifies the Interconnect_module (see 4.2.440) for the Interconnect_module_constraint_region.

Constraints:

WR1: The element_shape shall be a Csg_2d_shape.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_constraint_region
  SUBTYPE OF (non_feature_shape_element);
    keepout                                     : BOOLEAN;
    non_conformant_interconnect_module_design_object :
OPTIONAL SET [1:?] OF interconnect_module_design_object_select;
    SELF\non_feature_shape_element.scope       : interconnect_module;
    design_specific_purpose                       : ee_name;
    requirement                                : ee_requirement_occurrence;
    associated_stratum_extent                   : SET [1:?] OF inter_stratum_extent;
    constrained_design_object_category          :
interconnect_module_design_object_category;
INVERSE
    precedent_version_constraint_region         : SET [0:1] OF
      non_feature_shape_element_relationship for related_shape_element;
WHERE
  WR1: ('AP210_ARM.CSG_2D_SHAPE' IN
    TYPEOF(SELF\non_feature_shape_element.element_shape)) OR
    ('AP210_ARM.BOUND_VOLUME_SHAPE' IN
    TYPEOF(SELF\non_feature_shape_element.element_shape));
  WR2: NOT(SIZEOF(precedent_version_constraint_region) = 1) OR
    ('AP210_ARM.INTERCONNECT_MODULE_CONSTRAINT_REGION' IN
    TYPEOF(precedent_version_constraint_region\
      non_feature_shape_element_relationship.relatng_shape_element));
END_ENTITY; -- interconnect_module_constraint_region

( *
```

4.2.447 Interconnect_module_cutout_segment_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_cutout_segment_surface` is a type of `Interconnect_module_surface_feature` (see 4.2.462). An `Interconnect_module_cutout_segment_surface` identifies a segment of the cutout surface of the `Interconnect_module` (see 4.2.440) for fabrication and assembly purposes.

NOTE This data is usage view contextual data.

If a design view is provided and this member of `Interconnect_module_cutout_segment_surface` is specified by a member of `Stratum_concept_to_physical_usage_view_assignment` (see 4.2.850) in the role of `defined_part_feature`, a member of `Interconnect_module_cutout_segment` (see ??) or a member of `Plated_interconnect_module_cutout_segment` (see ??) shall be the only possibilities permitted to be specified for the role of `assigned_design_object` for that member of `Stratum_concept_to_physical_usage_view_assignment`.

The data associated with an `Interconnect_module_cutout_segment_surface` are the following:

- `composed_surface`;
- `end_vertex`;
- `start_vertex`.

4.2.447.1 composed_surface

The `composed_surface` specifies an `Interconnect_module_cutout_surface` for an `Interconnect_module_cutout_segment` (see ??).

4.2.447.2 end_vertex

The `end_vertex` specifies one role of an `Edge_segment_surface` for an `Interconnect_module_cutout_segment_surface`.

4.2.447.3 start_vertex

The `end_vertex` specifies one role of an `Edge_segment_surface` for an `Interconnect_module_cutout_segment_surface`.

Constraints:

WR1: The `end_vertex` shall not be the `start_vertex`.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_cutout_segment_surface
  SUBTYPE OF (interconnect_module_surface_feature);
    composed_surface : interconnect_module_cutout_surface;
    start_vertex      : edge_segment_vertex;
    end_vertex        : edge_segment_vertex;
WHERE
  WR1: start_vertex :&#60;&#62;; end_vertex;
END_ENTITY; -- interconnect_module_cutout_segment_surface

( *

```

4.2.448 Interconnect_module_cutout_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Interconnect_module_cutout_surface is a type of a Interconnect_module_surface_feature (see 4.2.462).

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_cutout_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_cutout_surface

( *

```

4.2.449 Interconnect_module_edge

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_edge is a type of Inter_stratum_feature (see 4.2.432). When this information is represented by a planar shape, the interpretation is to be that the planar shape represents a projection of the maximum extent of the Interconnect_module_edge onto the planar shape. The Interconnect_module_edge forms an outer physical boundary of the interconnect substrate.

Constraints:

WR1: Each member of Interconnect_module_edge shall only be an external feature of the Interconnect_module.

WR2: Any geometry representing a member of Interconnect_module_edge shall be interpreted by a receiving system as a closed curve or a closed volume, depending on the dimensionality of the geometry.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_edge
  SUPERTYPE OF (ONEOF(partially_plated_interconnect_module_edge,
    plated_interconnect_module_edge))
  SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- interconnect_module_edge

--tt tc1 wd9 modified
( *
```

4.2.450 Interconnect_module_edge_segment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_edge_segment is a type of Inter_stratum_feature (see 4.2.432). The Interconnect_module_edge_segment is an entity with a uniquely specified shape that makes up part of the edge of an interconnect module.

EXAMPLE 1 Usually each side of a four-sided board is an Interconnect_module_edge_segment, or if a board is a round board, one of the Interconnect_module_edge_segment's may be a notch for orientation purposes. The set of Interconnect_module_edge_segment's together make up the total edge.

The data associated with an Interconnect_module_edge_segment are the following:

- composed_edge;
- end_vertex;
- start_vertex.

4.2.450.1 composed_edge

The composed_edge specifies the role of the Interconnect_module_edge (see 4.2.449) for the Interconnect_module_edge_segment.

4.2.450.2 end_vertex

The end_vertex specifies one role of an Edge_segment_vertex for an Interconnect_module_edge_segment.

4.2.450.3 start_vertex

The start_vertex specifies one role of an Edge_segment_vertex for an Interconnect_module_edge_segment.

Constraints:

WR1: The end_vertex shall not be the start_vertex.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_edge_segment
  SUPERTYPE OF (plated_interconnect_module_edge_segment)
  SUBTYPE OF (inter_stratum_feature);
    composed_edge                : interconnect_module_edge;
    start_vertex                 : edge_segment_vertex;
    end_vertex                   : edge_segment_vertex;
    SELF\inter_stratum_feature.usage_definition :
      inter_stratum_feature_edge_segment_template;
WHERE
  WR1: start_vertex :&#60;&#62;: end_vertex;
END_ENTITY; -- interconnect_module_edge_segment

--tt tc1 wd9 modified
(

```

4.2.451 Interconnect_module_edge_segment_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_edge_segment_surface is a type of Interconnect_module_surface_feature (see 4.2.462). An Interconnect_module_edge_segment_surface identifies a segment of the edge surface of the Interconnect_module (see 4.2.440) for fabrication and assembly purposes.

NOTE This data is usage view contextual data.

If a design view is provided and this member of Interconnect_module_edge_segment_surface is specified by a member of Stratum_concept_to_physical_usage_view_assignment (see 4.2.850) in the role of defined_part_feature, a member of Interconnect_module_edge_segment (see 4.2.450) or a member of Plated_interconnect_module_edge_segment (see 4.2.689) shall be the only possibilities permitted to be specified for the role of assigned_design_object for that member of Stratum_concept_to_physical_usage_view_assignment.

The data associated with an Interconnect_module_edge_segment_surface are the following:

- composed_surface;
- end_vertex;
- start_vertex.

4.2.451.1 composed_surface

The composed_surface specifies an Interconnect_module_edge_surface for an Interconnect_module_edge_segment (see 4.2.450).

4.2.451.2 end_vertex

The end_vertex specifies one role of an Edge_segment_surface for an Interconnect_module_edge_segment_surface.

4.2.451.3 start_vertex

The end_vertex specifies one role of an Edge_segment_surface for an Interconnect_module_edge_segment_surface.

Constraints:

WR1: The end_vertex shall not be the start_vertex.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_edge_segment_surface
  SUBTYPE OF (interconnect_module_surface_feature);
    composed_surface : interconnect_module_edge_surface;
    start_vertex      : edge_segment_vertex;
    end_vertex        : edge_segment_vertex;
WHERE
  WR1: start_vertex :&#60;&#62;: end_vertex;
END_ENTITY; -- interconnect_module_edge_segment_surface

( *
```

4.2.452 Interconnect_module_edge_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_edge_surface is a type of Interconnect_module_surface_feature (see 4.2.462).

An Interconnect_module_edge_surface identifies the edge surface of the Interconnect_module (see 4.2.440) for fabrication and assembly purposes.

NOTE EXPRESS specification:

```
*)  
ENTITY interconnect_module_edge_surface  
  SUBTYPE OF (interconnect_module_surface_feature);  
END_ENTITY; -- interconnect_module_edge_surface  
  
--tt tc1 wd9 added  
(*
```

4.2.453 Interconnect_module_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_interface_terminal is a type of Interconnect_module_terminal (see 4.2.463).

NOTE EXPRESS specification:

```
*)  
ENTITY interconnect_module_interface_terminal  
  SUBTYPE OF (interconnect_module_terminal);  
END_ENTITY; -- interconnect_module_interface_terminal  
  
--tt tc1 wd9 added  
(*
```

4.2.454 Interconnect_module_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_join_terminal is a type of Interconnect_module_terminal (see 4.2.463).

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_join_terminal
  SUBTYPE OF (interconnect_module_terminal);
INVERSE
--tt tc1 wd9 modified
defining_terminal : component_feature_to_physical_usage_view_assignment for
  defined_part_feature;
WHERE
  WR1: SIZEOF([
    'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_JOIN_TERMINAL',
    'AP210_ARM.VIA_TERMINAL',
    'AP210_ARM.PRINTED_COMPONENT_JOIN_TERMINAL',
    'AP210_ARM.CONDUCTIVE_INTERCONNECT_ELEMENT_TERMINAL',
    'AP210_ARM.LAND_JOIN_TERMINAL',
    'AP210_ARM.NON_FUNCTIONAL_LAND_JOIN_TERMINAL'] *
    TYPEOF(defining_terminal.assigned_terminal)) = 1;
END_ENTITY; -- interconnect_module_join_terminal

--tt tc1 wd8 added
(*

```

4.2.455 Interconnect_module_macro_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Interconnect_module_macro_component is a type of Assembly_component (see 4.2.34).

The data associated with an Interconnect_module_macro_component are the following:

- access_mechanisms;
- usage_definition;
- usage_view_definition.

4.2.455.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Interconnect_module_macro_component is dependent on the existence of the Interconnect_module_component_terminal (see 4.2.445) that specifies the Interconnect_module_macro_component as its associated_component. There shall be two or more Interconnect_module_component_terminal for an Interconnect_module_macro_component.

4.2.455.2 usage_definition

The usage_definition specifies the role of an Interconnect_module (see 4.2.440) for the Interconnect_module_macro_component.

4.2.455.3 usage_view_definition

The usage_view_definition is a derived role of an Interconnect_module_usage_view (see 4.2.465) which specifies the specifies the Assembly_module_usage_view that is identified via the path SELF\assembly_component.usage_definition\physical_unit_design_view.usage_view.

Constraints:

WR1: The Interconnect_module_usage_view (see 4.2.465) referenced by the usage_view_definition shall be the same Interconnect_module_usage_view referenced by the associated_definition which is referenced by the definition which is referenced by the access_mechanisms.

NOTE EXPRESS specification:

*)

ENTITY interconnect_module_macro_component

--tt tc1 wd9 modified

SUBTYPE OF (interconnect_module_component);

--tt tc1 wd9 modified

design_definition : interconnect_module;

INVERSE

join_access_mechanisms : SET [1:?] OF

interconnect_module_macro_component_join_terminal FOR
associated_component;

WHERE

WR1: SIZEOF (QUERY(immcjt <* join_access_mechanisms |
immcjt\component_feature.definition\part_feature.associated_definition :<
design_definition\physical_unit_design_view.usage_view)) = 0;

END_ENTITY; -- interconnect_module_macro_component

--tt tc1 wd9 added

(*

4.2.456 Interconnect_module_macro_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_macro_component_join_terminal is a type of Interconnect_module_component_terminal (see 4.2.445).

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_macro_component_join_terminal
  SUBTYPE OF (interconnect_module_component_terminal);
  SELF\component_feature.definition      : interconnect_module_join_terminal;
  SELF\component_feature.associated_component : interconnect_module_macro_component;
END_ENTITY; -- interconnect_module_macro_component_join_terminal

( *

```

4.2.457 Interconnect_module_primary_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_primary_surface is a type of Interconnect_module_surface_feature (see 4.2.462). An Interconnect_module_primary_surface identifies the primary surface of the Interconnect_module (see 4.2.440) for fabrication and assembly purposes. A member of Stratum_concept_to_physical_usage_view_assignment (see 4.2.850) shall be populated in order to associate this object to the primary Stratum (see 4.2.845) if a design view of the substrate is included in the data. An Interconnect_module_primary_surface is an approximation of the real primary surface.

NOTE It is not usually the case that the physical material visible on the surface of the substrate or pcb is the same material as that of the primary Stratum. The level of information provided by this object is usually sufficient for assembly purposes, but will be in error by the thickness of the Stratum above the primary Stratum. If more accuracy is desired, the population of Adjacent_stratum_surface_definition (see 4.2.5) may be queried to determine the Stratum that is specified by only the precedent_surface attribute. That Stratum will be the material of at least part of the surface of the substrate. Full three-dimensional representation capability and location of individual Interconnect_module_component_terminal (see 4.2.445) associated by Assembly_joint (see 4.2.41) to the component feature being assembled to the substrate is required to represent exactly the mating conditions.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_primary_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_primary_surface

( *

```

4.2.458 Interconnect_module_product_assembly_view_fabrication_joint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_product_assembly_view_fabrication_joint` is a type of `Managed_design_object` (see 4.2.524). It specifies the bond among `Interconnect_module_component_stratum_based_terminal` (see 4.2.443) which is created during an assembly process. The detailed process description of how to create the bond is not included.

NOTE This Application Object is intended to support a limited view of an interconnect product where the vertical or sequential arrangement of intermediate assemblies is of interest, and the detailed information available in `Stratum` (see 4.2.845) is not of interest.

The data associated with an `Interconnect_module_product_assembly_view_fabrication_joint` are the following:

— `joined_terminal`.

4.2.458.1 `joined_terminal`

A `joined_terminal` specifies one or more `Interconnect_module_component_stratum_based_terminal` (see 4.2.443) for the `Interconnect_module_product_assembly_view_fabrication_joint`.

NOTE EXPRESS specification:

```
*)
ENTITY interconnect_module_product_assembly_view_fabrication_joint
  SUBTYPE OF (managed_design_object);
    joined_terminal :
SET [1:?] OF interconnect_module_component_stratum_based_terminal;
END_ENTITY; -- interconnect_module_product_assembly_view_fabrication_joint

( *
```

4.2.459 `Interconnect_module_secondary_surface`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_secondary_surface` is a type of `Interconnect_module_surface_feature` (see 4.2.462). An `Interconnect_module_secondary_surface` identifies the secondary surface of the `Interconnect_module` (see 4.2.440) for fabrication and assembly purposes. If a design view is populated, and correlation to a `Stratum` (see 4.2.845) is provided, the `Stratum` associated with this member of `Interconnect_module_primary_surface` (see 4.2.457) shall be the last `Stratum` in the design.

NOTE The `Stratum_concept_to_physical_usage_view_assignment` (see 4.2.850), `Adjacent_stratum_surface_definition` (see 4.2.5), and `Stratum_surface` (see 4.2.868) provide supporting data needed to resolve the position of the `Stratum` in the design. Because the secondary surface is derived structurally from the primary `Stratum`, there is no approximation made in the description.

NOTE EXPRESS specification:

```
*)
ENTITY interconnect_module_secondary_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_secondary_surface

( *
```

4.2.460 Interconnect_module_stratum_assembly_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_stratum_assembly_relationship is a type of Managed_design_object (see 4.2.524). An Interconnect_module_stratum_assembly_relationship is the composition association between an Interconnect_module (see 4.2.440) and the Stratum (see 4.2.845) helping to compose it.

The data associated with an Interconnect_module_stratum_assembly_relationship are the following:

- assembly;
- component;
- reference_designation;
- stratum_3d_position;
- stratum_planar_position.

4.2.460.1 assembly

The assembly specifies the role of the Interconnect_module (see 4.2.440) for the Interconnect_module_stratum_assembly_relationship. There shall be one or more Interconnect_module for the Interconnect_module_stratum_assembly_relationship.

4.2.460.2 component

The component specifies the role of the Stratum (see 4.2.845) for the Interconnect_module_stratum_assembly_relationship.

4.2.460.3 reference_designation

The reference_designation specifies the string for the human-interpretable identifier for the Interconnect_module_stratum_assembly_relationship.

4.2.460.4 stratum_3d_position

The stratum_3d_position is an inverse relationship that specifies the Stratum_interconnect_module_-3d_position (see 4.2.863) that specifies the Interconnect_module_stratum_assembly_relationship as its stratum_component. There shall be no more than one Stratum_interconnect_module_3d_position for an Interconnect_module_stratum_assembly_relationship.

4.2.460.5 stratum_planar_position

The stratum_planar_position is an inverse relationship that specifies the Stratum_interconnect_module_-planar_position (see 4.2.864) that specifies the Interconnect_module_stratum_assembly_relationship as its stratum_component. There shall be no more than one Stratum_interconnect_module_planar_position for an Interconnect_module_stratum_assembly_relationship.

Constraints:

UR1: The combination of the assembly and the reference_designation shall be unique within a population of Interconnect_module_stratum_assembly_relationship.

WR1: The members of assembly shall all be for the same product.

WR2: Each member of assembly shall be for a separate product version.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_stratum_assembly_relationship
  SUBTYPE OF (managed_design_object);
    assembly          : interconnect_module;
    component         : stratum;
    reference_designation : ee_name;
  INVERSE
    stratum_planar_position : SET [0:1] OF
                                stratum_interconnect_module_planar_position FOR
                                stratum_component;
    stratum_3d_position    : SET [0:1] OF
                                stratum_interconnect_module_3d_position FOR
                                stratum_component;
  UNIQUE
    UR1 : assembly, reference_designation;
END_ENTITY; -- interconnect_module_stratum_assembly_relationship

```

(*

4.2.461 Interconnect_module_stratum_based_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_stratum_based_terminal` is a type of `Interconnect_module_terminal` (see 4.2.463). An `Interconnect_module_stratum_based_terminal` identifies both the design object and the stratum surface related to the object in the design organization. This allows the use of the terminal in a component instance where surfaces are joined. It also supports placement of embedded components. An `Interconnect_module_stratum_based_terminal` includes the identification of a reference surface to support the sequence of application of an embedded component or of material buildup.

NOTE The `Interconnect_module_stratum_based_terminal` is an artifact that is defined in the design organization but generally will be assigned to only those lands or `non_functional_lands` that are intended to be penetrated with plated passages in the final product. This is only used in the case where sequential stackups are defined, and is derived from knowledge of the substrate design and intended lamination sequence. The other usage is for support of embedded components being connected into the succeeding stratum.

Constraints:

WR1: An `Interconnect_module_stratum_based_terminal` shall not be an assignment of an `Inter_stratum_feature` by a member of `Stratum_concept_to_physical_usage_view_assignment`.

WR2: An `Interconnect_module_stratum_based_terminal` shall be an assignment of a `Land` or `Non_functional_land` in the case where a member of `Stratum_concept_to_physical_usage_view_assignment` references the `Interconnect_module_stratum_based_terminal`.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_stratum_based_terminal
  SUBTYPE OF (interconnect_module_terminal);
WHERE
  WR1: SIZEOF(QUERY( imtscr &#60;* USEDIN(SELF,
    'AP210_ARM.' +
    'INTERCONNECT_MODULE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP.' +
    'RELATED_SHAPE_ELEMENT') |
    NOT (SIZEOF([ 'AP210_ARM.' +
    'INTERCONNECT_MODULE_PRIMARY_SURFACE',
    'AP210_ARM.' +
    'INTERCONNECT_MODULE_SECONDARY_SURFACE'] *
      TYPEOF(imtscr.relying_shape_element)) = 1 )
    )) = 0;
  WR2: SIZEOF(USEDIN(SELF,
    'AP210_ARM.' +
    'INTERCONNECT_MODULE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP.' +
    'RELATED_SHAPE_ELEMENT')) = 1;
  WR3: SIZEOF([ 'AP210_ARM.LAND',
    'AP210_ARM.NON_FUNCTIONAL_LAND'] *
    TYPEOF(SELF\part_feature.design_view_definition_component_feature)) = 1;

END_ENTITY; -- interconnect_module_stratum_based_terminal

( *
```

4.2.462 Interconnect_module_surface_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_surface_feature is a type of Part_feature (see 4.2.609). An Interconnect_module_surface_feature is one of an Interconnect_module_edge_segment_surface (see 4.2.451), an Interconnect_module_primary_surface (see 4.2.457), or an Interconnect_module_secondary_surface (see 4.2.459). An Interconnect_module_surface_feature identifies a surface of the Interconnect_module (see 4.2.440) for fabrication and assembly purposes, particular to support Component_2d_location (see 4.2.108).

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_surface_feature
  ABSTRACT SUPERTYPE OF (ONEOF (
    interconnect_module_cavity_surface,
    interconnect_module_cutout_segment_surface,
    interconnect_module_cutout_surface,
    interconnect_module_secondary_surface,
    interconnect_module_primary_surface,
    interconnect_module_edge_surface,
    interconnect_module_edge_segment_surface))
  SUBTYPE OF (part_feature);
  SELF\part_feature.associated_definition : interconnect_module_usage_view;
END_ENTITY; -- interconnect_module_surface_feature

( *

```

4.2.463 Interconnect_module_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interconnect_module_terminal` is a type of `Part_terminal` (see 4.2.630). The `Interconnect_module_terminal` shall be one of either an `Interconnect_module_interface_terminal` (see 4.2.453) or an `Interconnect_module_join_terminal` (see 4.2.454). The `Interconnect_module_terminal` identification is provided by the name attribute inherited from `Shape_element` (see 4.2.810). The string may include a substring that indicates to a person the connector the `Interconnect_module_terminal` is associated with. An `Interconnect_module_terminal` may be referenced by a member of `Part_connected_terminals_definition` (see 4.2.605) to identify a connection requirement for the `Interconnect_module`.

NOTE The `Interconnect_module_terminal` may be an `Interconnect_module_stratum_based_terminal` (see 4.2.461) but this fact is not structurally captured in this application object due to the need to require that the `Interconnect_module_terminal` shall be one of the two specified subtypes.

NOTE This part of ISO 10303 does not explicitly model, in a usage view, the relationship of a printed connector to an `Interconnect_module` in addition to the `associated_definition` relationship provided in this Application Object.

EXAMPLE 1 Figure 52 illustrates a detail of a usage view that shows an `Interconnect_module_terminal` that is a part of a edge finger connector.

The data associated with an `Interconnect_module_terminal` are the following:

- `associated_definition`;
- `feature_shape` ;
- `terminal_connection_zone` .

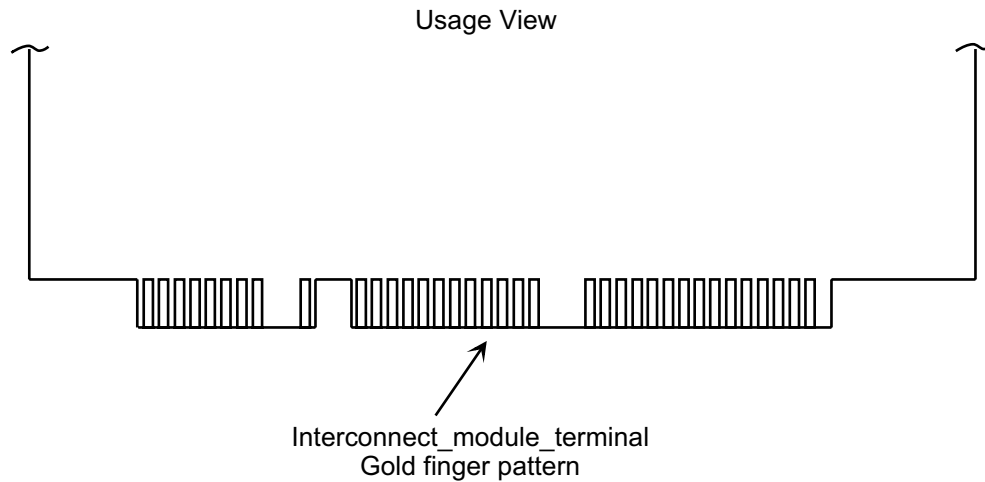


Figure 52 – Interconnect_module_terminal

4.2.463.1 associated_definition

The `associated_definition` specifies the role of the `Interconnect_module_usage_view` (see 4.2.465) for the `Interconnect_module_terminal`.

4.2.463.2 feature_shape

The `feature_shape` specifies the role of the `Feature_shape_occurrence` (see ??) for the `Interconnect_module_terminal`. There shall be one or more `Feature_shape_occurrence` for an `Interconnect_module_terminal`.

4.2.463.3 terminal_connection_zone

The `terminal_connection_zone` specifies the role of the `Connection_zone` (see 4.2.185) for the `Interconnect_module_terminal`. There may be more than one `Connection_zone` for an `Interconnect_module_terminal`. The `terminal_connection_zone` is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the data shall be provided.

NOTE The `Feature_shape_occurrence` (see ??) specified by the terminal or mounting feature may be the same `Feature_shape_occurrence` that is specified by the associated `Connection_zone` (see 4.2.185). In this case the whole terminal is the `Connection_zone`.

NOTE EXPRESS specification:

```

*)
ENTITY interconnect_module_terminal
  SUPERTYPE OF (ONEOF(
    interconnect_module_interface_terminal,
    interconnect_module_join_terminal))
  SUBTYPE OF (part_terminal);
  terminal_connection_zone : OPTIONAL SET [1:?] OF connection_zone;
  SELF\part_feature.associated_definition : interconnect_module_usage_view;
INVERSE
  SELF\part_feature.feature_shape : SET [1:?] OF usage_concept_usage_relationship;
  associated_usage;
END_ENTITY; -- interconnect_module_terminal
( *

```

4.2.464 Interconnect_module_terminal_surface_constituent_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_terminal_surface_constituent_relationship is a type of Shape_element_constituent_relationship (see 4.2.813).

The data associated with an Interconnect_module_terminal_surface_constituent_relationship are the following:

- relating_shape_element;
- related_shape_element.

4.2.464.1 relating_shape_element

The relating_shape_element specifies the role of the Interconnect_module_surface_feature (see 4.2.462) for the Interconnect_module_terminal_surface_constituent_relationship.

4.2.464.2 related_shape_element

The related_shape_element specifies the role of the Interconnect_module_terminal (see 4.2.463) for the Interconnect_module_terminal_surface_constituent_relationship.

NOTE EXPRESS specification:

*)

```
ENTITY interconnect_module_terminal_surface_constituent_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relatng_shape_element : interconnect_module_surface_fea
  SELF\shape_element_relationship.related_shape_element : interconnect_module_terminal;
END_ENTITY; -- interconnect_module_terminal_surface_constituent_relationship
```

(*

4.2.465 Interconnect_module_usage_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interconnect_module_usage_view is a type of Physical_unit_usage_view (see 4.2.672). An Interconnect_module_usage_view may be a Pcb_usage_view (see 4.2.643). Either the maximum_thickness_over_dielectric_requirement or the maximum_thickness_over_metal_requirement shall be provided. If both the maximum_thickness_over_dielectric_requirement and the maximum_thickness_over_metal_requirement are provided, the maximum_thickness_over_metal_requirement shall be greater than the maximum_thickness_over_dielectric_requirement.

The data associated with an Interconnect_module_usage_view are the following:

- implemented_function;
- located_thickness_requirement;
- maximum_thickness_over_dielectric_requirement;
- maximum_thickness_over_metal_requirement;
- measurement_condition;
- minimum_thickness_over_dielectric_requirement;
- minimum_thickness_over_metal_requirement.

4.2.465.1 implemented_function

The implemented_function specifies the role of the Functional_unit_usage_view (see 4.2.410) for the Interconnect_module_usage_view. The implemented_function need not be specified for a particular Interconnect_module_usage_view.

4.2.465.2 located_thickness_requirement

The located_thickness_requirement specifies the role of the Located_interconnect_module_thickness_requirement (see 4.2.519) for the Interconnect_module_usage_view. A Located_interconnect_module_thickness_requirement need not be specified for an Interconnect_module_usage_view.

4.2.465.3 maximum_thickness_over_dielectric_requirement

The maximum_thickness_over_dielectric_requirement specifies the role of the Length_data_element (see 4.2.511) for the Interconnect_module_usage_view. The maximum_thickness_over_dielectric_requirement need not be specified for a particular Interconnect_module_usage_view. The maximum_thickness_over_dielectric_requirement references a base dielectric and not final finish dielectrics.

4.2.465.4 maximum_thickness_over_metal_requirement

The maximum_thickness_over_metal_requirement specifies the role of the Length_data_element (see 4.2.511) for the Interconnect_module_usage_view. The maximum_thickness_over_metal_requirement need not be specified for a particular Interconnect_module_usage_view.

4.2.465.5 measurement_condition

The measurement_condition specifies the role of the Restraint_condition (see 4.2.774) for the Interconnect_module_usage_view. The measurement_condition need not be specified for a particular Interconnect_module_usage_view.

4.2.465.6 minimum_thickness_over_dielectric_requirement

The minimum_thickness_over_dielectric_requirement specifies the role of the Length_data_element (see 4.2.511) for the Interconnect_module_usage_view. The minimum_thickness_over_dielectric_requirement need not be specified for a particular Interconnect_module_usage_view. The minimum_thickness_over_dielectric_requirement references a base dielectric and not final finish dielectrics.

4.2.465.7 minimum_thickness_over_metal_requirement

The minimum_thickness_over_metal_requirement specifies the role of the Length_data_element (see 4.2.511) for the Interconnect_module_usage_view. The minimum_thickness_over_metal_requirement need not be specified for a particular Interconnect_module_usage_view.

Constraints:

WR1: At least one thickness property shall be provided.

NOTE EXPRESS specification:

*)

```

ENTITY interconnect_module_usage_view
  SUPERTYPE OF (pcb_usage_view)
  SUBTYPE OF (physical_unit_usage_view);
    minimum_thickness_over_metal_requirement      : OPTIONAL length_data_element;
    maximum_thickness_over_metal_requirement      : OPTIONAL length_data_element;
    minimum_thickness_over_dielectric_requirement : OPTIONAL length_data_element;
    maximum_thickness_over_dielectric_requirement : OPTIONAL length_data_element;
    measurement_condition                        : OPTIONAL restraint_condition;
    located_thickness_requirement                :
OPTIONAL SET [1:?] OF located_interconnect_module_thickness_requirement;
    implemented_function                        : OPTIONAL functional_unit_usage_view;
WHERE
  WR1: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'INTERCONNECT_MODULE_PRIMARY_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR2: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'INTERCONNECT_MODULE_SECONDARY_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR3: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') |
    ('AP210_ARM.' +
    'INTERCONNECT_MODULE_EDGE_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR4: EXISTS(maximum_thickness_over_metal_requirement) OR
    EXISTS(maximum_thickness_over_dielectric_requirement);
  WR5: NOT(EXISTS(minimum_thickness_over_metal_requirement)) OR
    (minimum_thickness_over_metal_requirement\data_element.measure_value &#62; 0.0);
  WR6: NOT(EXISTS(maximum_thickness_over_metal_requirement)) OR
    (maximum_thickness_over_metal_requirement\data_element.measure_value &#62; 0.0);
  WR7: NOT(EXISTS(minimum_thickness_over_dielectric_requirement)) OR
    (minimum_thickness_over_dielectric_requirement\data_element.measure_value &#62; 0.0);
  WR8: NOT(EXISTS(maximum_thickness_over_dielectric_requirement)) OR
    (maximum_thickness_over_dielectric_requirement\data_element.measure_value &#62; 0.0);
END_ENTITY; -- interconnect_module_usage_view
(*)

```

4.2.466 Interface_access_material_removal_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interface_access_material_removal_laminate_component is a type of Material_removal_laminate_component (see 4.2.535) that indicates that material is to be removed in the realization of access to the Laminate_component_interface_terminal.

EXAMPLE 1 If a passage in a silkscreen layer is necessary to gain access to a `Laminate_component_interface_terminal` and it is desire to record the reason for the passage, this Application Object provides that capability.

The data associated with an `Interface_access_material_removal_laminate_component` are the following:

— `accessed_interface`.

4.2.466.1 `accessed_interface`

The `accessed_interface` specifies the role of the `Laminate_component_interface_terminal` (see 4.2.485) for the `Interface_access_material_removal_laminate_component`.

NOTE EXPRESS specification:

```
*)
ENTITY interface_access_material_removal_laminate_component
  SUBTYPE OF (material_removal_laminate_component);
  accessed_interface : laminate_component_interface_terminal;
END_ENTITY; -- interface_access_material_removal_laminate_component

( *
```

4.2.467 `Interface_access_stratum_feature_template_component`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interface_access_stratum_feature_template_component` is a type of `Stratum_feature_template_component` (see 4.2.862) provided to provide access to a `Laminate_component_interface_terminal` where auxiliary processes would prohibit direct access to the terminal and where it is necessary to record the explicit relationship.

NOTE A solderpaste may cover the terminal and in some cases it is necessary to denote the fact that the terminal is below the solderpaste. If it is not necessary to explicitly identify the fact that the solderpaste covers the terminal, this Application Object would not be populated. Those cases would occur where geometry queries are sufficient. The `Interface_access_stratum_feature_template_component` is not on the same `Stratum` as the terminal being accessed.

NOTE EXPRESS specification:

```
*)
ENTITY interface_access_stratum_feature_template_component
  SUBTYPE OF (stratum_feature_template_component);
  accessed_interface : laminate_component_interface_terminal;
END_ENTITY; -- interface_access_stratum_feature_template_component

( *
```

4.2.468 Interface_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interface_component is a type of Assembly_component (see 4.2.34) used to provide an identified interface in the assembly.

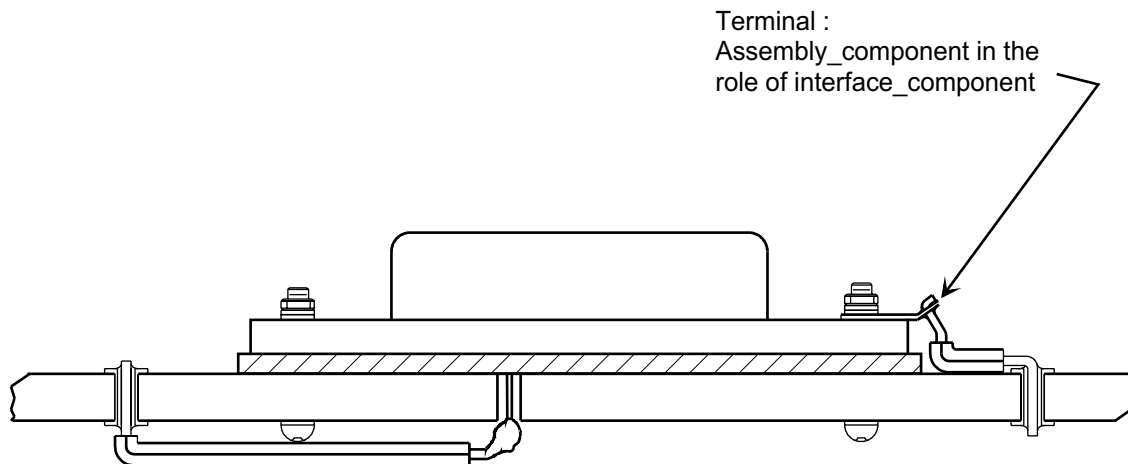
EXAMPLE 1 Figure 53 illustrates an Interface_component and its role in the assembly, and that there may be multiple Interface_component in sequence.

NOTE The interface provided by the Interface_component is usually included in the design to provide an interface between the component providing the primary interconnect functionality and other components or to the next level of assembly.

EXAMPLE 2 A bond wire that attaches a die terminal to a substrate terminal is an Interface_component. The logic behind this is the the bond wire is an electrical interface between the die instance and the substrate instance, and that it is separate from the die and from the substrate. If the routing is unique, the bond wire may also be a Routed_interconnect_component. In the case of a bond wire, only one terminal would be a Component_interface_-terminal. The second terminal would be an Interconnect_component_join_terminal.

The data associated with an Interface_component are the following:

— access_mechanism.



Note - the insulated bus wires are, individually, Assembly_component that is a Routed_interconnect_component and is also an Interface_component

Figure 53 – Interface_component

4.2.468.1 access_mechanism

The `access_mechanism` is an inverse relationship that specifies that the existence of the `Interface_component` is dependent on the existence of the `Component_interface_terminal` (see 4.2.125) that specifies the `Interface_component` as its `associated_component`. There shall be one or more `Component_interface_terminal` for an `Interface_component`.

NOTE EXPRESS specification:

```
*)
ENTITY interface_component
  SUPERTYPE OF (packaged_connector_component)
  SUBTYPE OF (physical_component);
  INVERSE
    access_mechanism : SET [1:?] OF
                                component_interface_terminal FOR associated_component;
END_ENTITY; -- interface_component

(*
```

4.2.469 Interface_component_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interface_component_allocation` is a type of `Managed_design_object` (see 4.2.524). An `Interface_component_allocation` is the assignment relationship between an `Interface_component` (see 4.2.468) on an assembly and the `Mating_connector_component` (see 4.2.538) to which the `Interface_component` must mechanically and electrically mate in the next assembly level.

The data associated with an `Interface_component_allocation` are the following:

- `mating_requirement`;
- `mating_solution`.

4.2.469.1 mating_requirement

The `mating_requirement` specifies the role of the `Mating_connector_component` (see 4.2.538) for the `Interface_component_allocation`.

4.2.469.2 mating_solution

The `mating_solution` specifies the role of the `Interface_component` (see 4.2.468) for the `Interface_component_allocation`.

NOTE EXPRESS specification:

```

*)
ENTITY interface_component_allocation
  SUBTYPE OF (managed_design_object);
    mating_solution      : interface_component;
    mating_requirement : mating_connector_component;
END_ENTITY; -- interface_component_allocation

( *
```

4.2.470 Interface_mounted_join

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interface_mounted_join is a type of Managed_design_object (see 4.2.524). An Interface_mounted_join is the relationship that specifies the connectivity between a terminal on a component and the terminal on a socket in the assembly that directly mates with that component terminal.

EXAMPLE 1 An example of this relation is pin 1, a Component_terminal (see 4.2.137), on U1, an integrated circuit and a component, that is connected to pin 1, on XU1, a socket and an Interface_component (see 4.2.468), when the IC is inserted into the socket in the assembly.

NOTE This Application object is not referencing the terminal on the socket that mates with the base interconnect substrate. That is referenced by a member of Assembly_joint.

The data associated with an Interface_mounted_join are the following:

- access_to_interface_mounted_component;
- connection_to_interconnect_module_component.

4.2.470.1 access_to_interface_mounted_component

The access_to_interface_mounted_component specifies the Component_terminal (see 4.2.137) of the Assembly_component (see 4.2.34) being interfaced to the assembly by the Interface_component (see 4.2.468) specified by connection_to_interconnect_module_component.

4.2.470.2 connection_to_interconnect_module_component

The connection_to_interconnect_module_component specifies the Component_interface_terminal (see 4.2.125) that mates with the Component_terminal (see 4.2.137) specified by access_to_interface_mounted_component. The Component_interface_terminal is a terminal of the Interface_component (see 4.2.468).

Constraints:

WR1: A `Physical_connectivity_definition` shall not specify a `Component_terminal` that participates in an `Interface_mounted_join` relationship as `access_to_interface_mounted_component` if that `Physical_connectivity_definition` specifies the `Component_terminal` of the associated `Interface_component` that completes connectivity to the board from the `interface_mounted_component`.

WR2: All members of `Component_terminal` specified by associated `terminals` shall be in the same design view.

NOTE It is inappropriate to specify both the terminal on an IC plugged into a socket and the socket terminal in the same nodelist. It is unnecessary to specify the interface terminal on a socket in a nodelist, as the socket terminal is referenced by a member of `Interface_mounted_join` (see 4.2.470) that is referenced by a member of `Physical_node_requirement_to_implementing_component_allocation` (see 4.2.663) that references the original nodelist.

WR3: Both components whose terminals are being joined shall be for the same version of the Product.

NOTE EXPRESS specification:

```

*)
ENTITY interface_mounted_join
  SUBTYPE OF (managed_design_object);
  --tt tc1 wd13 modified
    connection_to_interconnect_module_component : packaged_connector_component_interface_t
  --tt tc1 wd12 modified
    access_to_interface_mounted_component      : physical_component_terminal;
END_ENTITY; -- interface_mounted_join

( *
```

4.2.471 Interface_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Interface_requirement_occurrence` is a type of `Design_requirement_occurrence` (see 4.2.279) in which the assembly interface to the next higher assembly is specified.

The data associated with an `Interface_requirement_occurrence` are the following:

— `interface_to_next_higher_assembly`.

4.2.471.1 interface_to_next_higher_assembly

The `interface_to_next_higher_assembly` specifies the role of the `Next_higher_assembly_interface_requirement` (see 4.2.555) for the `Interface_requirement_occurrence`. This specifies the component in the next assembly that is the interface.

NOTE EXPRESS specification:

```
*)
ENTITY interface_requirement_occurrence
  SUBTYPE OF (design_requirement_occurrence);
  interface_to_next_higher_assembly : next_higher_assembly_interface_requirement;
END_ENTITY; -- interface_requirement_occurrence

( *
```

4.2.472 Interface_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interface_specification is a type of Design_specification (see 4.2.280) that describes the characteristics and constraints necessary for correct interoperation between specified units.

NOTE EXPRESS specification:

```
*)
ENTITY interface_specification
  SUBTYPE OF (design_specification);
END_ENTITY; -- interface_specification

( *
```

4.2.473 Interfaced_group_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interfaced_group_component is a type of Assembly_group_component (see 4.2.40). An Interfaced_group_component includes one or more Interface_component (see 4.2.468) that interfaces a subset of that group of components to the assembly.

EXAMPLE 1 A component and a socket are both assigned to an Interfaced_group_component. The socket interfaces the other component to the assembly. Both the component and the socket are separately referenced and placed in the assembly.

The data associated with an Interfaced_group_component are the following:

— group_interface_component.

4.2.473.1 group_interface_component

The group_interface_component specifies the Interface_component (see 4.2.468) that interfaces the Interfaced_group_component to the assembly. There shall be one or more Interface_component for an Interfaced_group_component.

NOTE EXPRESS specification:

```

*)
ENTITY interfaced_group_component
  SUBTYPE OF (assembly_group_component);
  group_interface_component : SET [1:?] OF interface_component;
END_ENTITY; -- interfaced_group_component

( *
```

4.2.474 Interfacial_connection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Interfacial_connection is a type of Via (see 4.2.947) that has both ends visible in the printed circuit board. Interfacial_connection is defined in IEC 60050-541.

EXAMPLE 1 Figure 54 illustrates an Interfacial_connection in a cross-section view of a substrate.

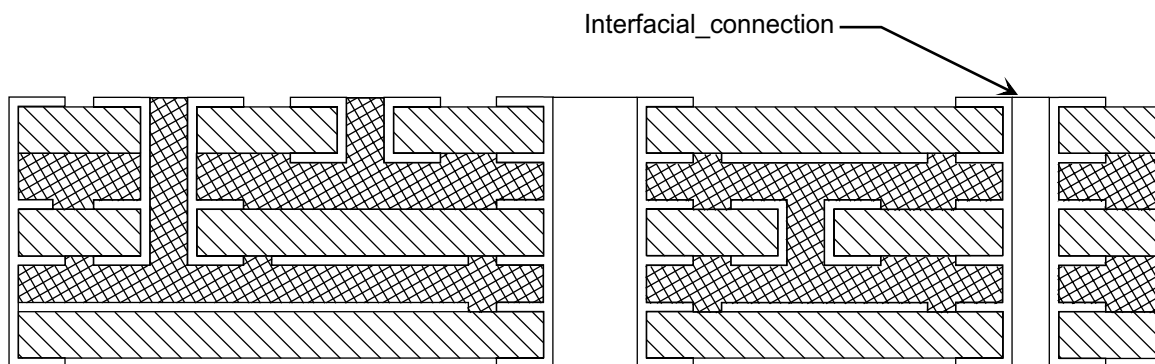


Figure 54 – Interfacial_connection

NOTE EXPRESS specification:

```
*)
ENTITY interfacial_connection
  SUBTYPE OF (via);
END_ENTITY; -- interfacial_connection

( *
```

4.2.475 Internal_probe_access_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Internal_probe_access_area is a type of Probe_access_area (see 4.2.739) included in the design for access to internal conductors for probing purposes.

NOTE This is included explicitly because probe points are not usually included in a functional nodelist.

The data associated with an Internal_probe_access_area are the following:

— stratum_feature_implementation.

4.2.475.1 stratum_feature_implementation

The stratum_feature_implementation specifies the role of the Stratum_feature (see 4.2.851) for the Internal_probe_access_area. This Stratum_feature is included only to support the probe requirements and is not necessarily included in a functional nodelist.

Constraints:

WR1: Each member of Internal_probe_access_area shall be in the role of connected_probe for exactly one member of Inter_stratum_access.

WR2: The Stratum_feature specified by stratum_feature_implementation shall specify an external Stratum as a resident_stratum.

NOTE EXPRESS specification:

```
*)
ENTITY internal_probe_access_area
  SUBTYPE OF (probe_access_area);
  stratum_feature_implementation : stratum_feature;
END_ENTITY; -- internal_probe_access_area

( *
```

4.2.476 Internal_stratum_access

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Internal_stratum_access is a type of Managed_design_object (see 4.2.524). An Internal_stratum_access is a relationship between an Internal_probe_access_area (see 4.2.475) and a Via (see 4.2.947) that provides access to a Stratum_feature (see 4.2.851) on an internal Design_layer_stratum (see 4.2.275). This Application Object is part of a logical structure to identify the special usage of a Via in support of accessing internal conductors in the design.

The data associated with an Internal_stratum_access are the following:

- connected_probe;
- inter_stratum_feature.

4.2.476.1 connected_probe

The connected_probe specifies the Internal_probe_access_area (see 4.2.475) that provides probe access to a Stratum_feature (see 4.2.851) on an internal Design_layer_stratum (see 4.2.275) for the Internal_stratum_access.

4.2.476.2 inter_stratum_feature

The inter_stratum_feature specifies the Via (see 4.2.947) that provides electrical connectivity to the Stratum_feature (see 4.2.851) to be probed for the Internal_stratum_access. The external surface of the Via is the area the test probe mates with in a test setup.

Constraints:

UR1: The combination of the connected_probe and the inter_stratum_feature shall be unique within a population of Internal_stratum_accesses.

NOTE EXPRESS specification:

```

*)
ENTITY internal_stratum_access
  SUBTYPE OF (managed_design_object);
    connected_probe      : internal_probe_access_area;
    inter_stratum_feature : via;
  UNIQUE
    UR1 : connected_probe, inter_stratum_feature;
END_ENTITY; -- internal_stratum_access

--tt tc1 wd8 modified
( *
```

4.2.477 Intra_stratum_join_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Intra_stratum_join_relationship is a type of Join_relationship (see 4.2.479) in which all joins occur on one Design_layer_stratum (see 4.2.275).

The data associated with an Intra_stratum_join_relationship are the following:

- ordered_implementation;
- random_implementation.

4.2.477.1 ordered_implementation

The ordered_implementation is an inverse relationship that specifies the Conductor (see 4.2.176) that specifies the Intra_stratum_join_relationship as its implementation_requirement. There shall be no more than one Conductor for an Intra_stratum_join_relationship. The use of "ordered" in this context shall be interpreted as "having a distinct structure by design". It does not indicate a strictly linear sequence.

4.2.477.2 random_implementation

The random_implementation is an inverse relationship that specifies the Connected_filled_area (see 4.2.184) that specifies the Intra_stratum_join_relationship as its implementation_requirement. There shall be no more than one Connected_filled_area for an Intra_stratum_join_relationship. The use of "random" in this context shall be interpreted as "having no intrinsic structure by design".

Constraints:

WR1: All Layer_connection_points specified shall be on the same Design_layer_stratum.

WR2: At least one ordered_implementation or random_implementation shall exist for a population of Intra_stratum_join_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY intra_stratum_join_relationship
  SUBTYPE OF (join_relationship);
  points_to_be_connected : SET [2:?] OF layer_connection_point;
  INVERSE
    ordered_implementation : SET [0:1] OF conductor FOR implementation_requirement;
    random_implementation  : SET [0:1] OF connected_filled_area FOR
      implementation_requirement;
--tt tc1 wd9 added
WHERE
  WR1: (EXISTS(ordered_implementation)) XOR (EXISTS(random_implementation));
END_ENTITY; -- intra_stratum_join_relationship

( *

```

4.2.478 Item_restricted_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Item_restricted_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330). The Item_restricted_requirement_occurrence provides the ability for a design organization to track the dependency of a requirement on a specific design object.

EXAMPLE 1 Thermal requirements may, for example, be dependent on the processor module or the drive motor.

The data associated with an Item_restricted_requirement_occurrence are the following:

— basis.

4.2.478.1 basis

The basis specifies the role of one of an Assembly_component (see 4.2.34), a Component_external_reference (see 4.2.117), a Component_feature (see 4.2.118); or a Component_feature_external_reference (see 4.2.119); or one of an Ee_requirement_occurrence (see 4.2.330), a Stratum (see 4.2.845); or one of an Inter_stratum_feature (see 4.2.432), a Laminate_component (see 4.2.483), a Stratum_feature (see 4.2.851), or a Stratum_surface (see 4.2.868) for the Item_restricted_requirement_occurrence. The basis is the reference design object for the requirement.

NOTE EXPRESS specification:

```

*)
ENTITY item_restricted_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
    basis : restriction_basis_item;
END_ENTITY; -- item_restricted_requirement_occurrence

( *

```

4.2.479 Join_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Join_relationship is a type of Managed_design_object (see 4.2.524). A Join_relationship is either an Inter_stratum_join_relationship (see 4.2.437) or an Intra_stratum_join_relationship (see 4.2.477). A Join_relationship identifies a strict subset of the Layer_connection_point (see 4.2.499) identified by Physical_network (see 4.2.660) topology. This subset is then inherited by both types of Join_relationship to ensure topological integrity.

The data associated with a Join_relationship are the following:

- associated_layout;
- network_topology;
- points_to_be_connected.

4.2.479.1 associated_layout

The associated_layout specifies the Interconnect_module (see 4.2.440) that is the design in which the Join_relationship is contained.

4.2.479.2 network_topology

The network_topology is an inverse relationship that specifies that the existence of the Join_relationship is dependent on the existence of the Physical_network (see 4.2.660) that specifies the Join_relationship as its interconnect_module_network_topology.

4.2.479.3 points_to_be_connected

The points_to_be_connected specifies the role of the Layer_connection_point (see 4.2.499) for the Join_relationship. There shall be at least two Layer_connection_point in a Join_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY join_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (inter_stratum_join_relationship,
    intra_stratum_join_relationship,
    unrouted_join_relationship))
  SUBTYPE OF (managed_design_object);
  associated_layout : interconnect_module;
  INVERSE
    network_topology : physical_network FOR
      interconnect_module_network_topology_element;
END_ENTITY; -- join_relationship
( *

```

4.2.480 Join_two_physical_connectivity_definition_supporting_inter_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Join_two_physical_connectivity_definition_supporting_inter_stratum_feature is a type of Physical_network_supporting_inter_stratum_feature (see 4.2.661).

The data associated with a Join_two_physical_connectivity_definition_supporting_inter_stratum_feature are the following:

— joined_definitions.

4.2.480.1 joined_definitions

The joined_definitions specifies the role of a Physical_connectivity_definition (see 4.2.652) for the Join_two_physical_connectivity_definition_supporting_inter_stratum_feature. There shall be exactly two Physical_connectivity_definition for a Join_two_physical_connectivity_definition_supporting_inter_stratum_feature.

NOTE EXPRESS specification:

```

*)
ENTITY join_two_physical_connectivity_definition_supporting_inter_stratum_feature
  SUBTYPE OF (physical_network_supporting_inter_stratum_feature);
  joined_definitions : SET [2:2] OF physical_connectivity_definition;
END_ENTITY; -- join_two_physical_connectivity_definition_
  -- supporting_inter_stratum_feature

--tt tc1wd8 modified
--tt tc1wd2 correct interpretation error
( *

```

4.2.481 Join_two_physical_connectivity_definition_supporting_printed_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Join_two_physical_connectivity_definition_supporting_printed_component is a type of Printed_component (see 4.2.721). The Join_two_physical_connectivity_definition_supporting_printed_component is included specifically for cases in which analog and digital grounds are kept topologically and physically separate completely through the design process to minimize noise but need to be physically connected for reference voltage purposes.

NOTE This connections is usually required to be at one and only one location.

The data associated with a Join_two_physical_connectivity_definition_supporting_printed_component are the following:

— joined_definitions.

4.2.481.1 joined_definitions

The joined_definitions specify the roles of exactly two Physical_connectivity_definition (see 4.2.652) for the Join_two_physical_connectivity_definition_supporting_printed_component.

NOTE EXPRESS specification:

```
* )
ENTITY join_two_physical_connectivity_definition_supporting_printed_component
  SUBTYPE OF (conductive_interconnect_element_with_pre_defined_transitions);
  joined_definitions : SET [2:2] OF physical_connectivity_definition;
END_ENTITY; -- join_two_physical_connectivity_definition_supporting_printed_component

( *
```

4.2.482 Junction_vertex_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Junction_vertex_allocation` is a type of `Managed_design_object` (see 4.2.524). A `Junction_vertex_allocation` is the relationship that specifies one or more `Layer_connection_point` (see 4.2.499) to which a `Topological_junction` (see 4.2.928) is mapped during the layout routing process. A `Junction_vertex_allocation` is the result of an assignment process by a designer or a layout routing process that defines the location of `Inter_stratum_join_relationship` (see 4.2.437) or `Explicitly_located_layer_connection_point` (see 4.2.360). A `Junction_vertex_allocation` maps a node of a `Physical_network` which may be only partially ordered into a network description which is highly ordered using a tree. A `Junction_vertex_allocation` associates two `Ee_product_definitions`, one an `Interconnect_module` and the other a `Physical_unit_network_definition`. The `Physical_unit_network_definition` may be populated for the express purpose of representing the tree structure of the `Physical_network` in the `Interconnect_module`, and therefore the version of the design would be the same for both.

The data associated with a `Junction_vertex_allocation` are the following:

- `implementation_point`;
- `required_vertex`.

4.2.482.1 implementation_point

The `implementation_point` specifies either the `Inter_stratum_join_relationship` (see 4.2.437) or `Explicitly_located_layer_connection_point` (see 4.2.360) that, individually, maps the requirement that a junction participates in a connectivity onto points on conducting Stratum (see 4.2.845) for the `Junction_vertex_allocation`.

NOTE 1 This construct allows the `Junction_vertex_allocation` to consider an `Inter_stratum_join_relationship` to be a vertex.

NOTE 2 The `Junction_vertex_allocation` allows for the specification of the location of the connections between physical traces and connected areas. It is common practice for the point where a trace, via, pad, etc., connects with other features to be specified by the designer interactively during layout. This location may be constrained for fabrication, layout, or signal integrity purposes. The `Junction_vertex_allocation` specifies these relationships.

4.2.482.2 required_vertex

The `required_vertex` specifies the `Topological_junction` (see 4.2.928), as a vertex of a `Physical_connectivity_definition` (see 4.2.652), that is to be mapped to an `implementation_point` by the `Junction_vertex_allocation`.

Constraints:

UR1: The `required_vertex` shall be unique within a population of `Junction_vertex_allocations`.

WR1: The `implementation_point` and `required_vertex` shall be for the same product.

NOTE EXPRESS specification:

```

*)
ENTITY junction_vertex_allocation
  SUBTYPE OF (managed_design_object);
    implementation_point : trace_junction_or_via;
    required_vertex      : topological_junction;
  UNIQUE
    UR1 : required_vertex;
END_ENTITY; -- junction_vertex_allocation

( *
```

4.2.483 Laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_component is a type of Assembly_component (see 4.2.34). A Laminate_component may be one of a Conductive_interconnect_element (see 4.2.171), an Inter_stratum_feature (see 4.2.432), a Laminate_text_component (see 4.2.488), a Laminate_text_string_component (see 4.2.489), a Material_removal_laminate_component (see 4.2.535), a Probe_access_area (see 4.2.739), a Special_symbol_laminate_component (see 4.2.834) or a Stratum_feature_template_component (see 4.2.862), or additionally may be a Fiducial (see 4.2.373), or additionally may be a Physical_laminate_component (see 4.2.659). All types of Laminate_component are realized as part of the manufacturing process that produces the interconnect.

The data associated with a Laminate_component are the following:

- cad_layer;
- usage_definition.

NOTE Complex multi-layered components should be described by using the Assembly_group_component and Sub_assembly_component_relationship. These relationships will be created from occurrences of Stratum_feature_template_component or other Printed_component occurrences.

4.2.483.1 cad_layer

The cad_layer specifies the role of the External_definition (see 4.2.361) for the Laminate_component. The cad_layer need not be specified for a Laminate_component.

NOTE The cad_layer attribute is provided to support design archival within an enterprise.

4.2.483.2 usage_definition

The usage_definition specifies the role of the Part_template (see 4.2.620) for the Laminate_component.

NOTE EXPRESS specification:

```

*)
ENTITY laminate_component
  SUPERTYPE OF ((ONEOF (conductive_interconnect_element,
    material_removal_laminate_component,
    stratum_feature_template_component,
    probe_access_area, inter_stratum_feature,
    special_symbol_laminate_component, laminate_text_component,
    laminate_text_string_component) ANDOR fiducial) ANDOR
    physical_laminate_component)
  SUBTYPE OF (assembly_component);
  cad_layer : OPTIONAL external_definition;
  SELF\assembly_component.usage_definition : OPTIONAL part_template;
END_ENTITY; -- laminate_component
( *

```

4.2.484 Laminate_component_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_component_feature is a

The data associated with a Laminate_component_feature are the following:

— additional_data.

4.2.484.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY laminate_component_feature
  ABSTRACT SUPERTYPE OF (ONEOF (
    conductive_interconnect_element_terminal,
    laminate_component_interface_terminal,
    laminate_component_join_terminal,
    via_terminal))
  SUBTYPE OF (component_feature);
END_ENTITY; -- laminate_component_feature

--tt tc1 wd12 added
( *

```

4.2.485 Laminate_component_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_component_interface_terminal is a

The data associated with a Laminate_component_interface_terminal are the following:

— additional_data.

4.2.485.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY laminate_component_interface_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (
    non_functional_land_interface_terminal,
    printed_connector_component_interface_terminal,
    component_termination_passage_interface_terminal,
    land_interface_terminal))
  SUBTYPE OF (laminate_component_feature);
END_ENTITY; -- laminate_component_interface_terminal

--tt tc1 wd12 added
( *
```

4.2.486 Laminate_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_component_join_terminal is a Laminate_component_feature and a Component_terminal.

The data associated with a Laminate_component_join_terminal are the following:

— additional_data.

4.2.486.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY laminate_component_join_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (
    component_termination_passage_join_terminal,
    land_join_terminal,
    non_functional_land_join_terminal,
    embedded_component_terminal,
    printed_component_join_terminal))
  SUBTYPE OF (laminate_component_feature, component_terminal);
INVERSE
  node_implementation : SET [0:1] OF physical_network FOR
    connectivity_requirement_element;
END_ENTITY; -- laminate_component_join_terminal

( *

```

4.2.487 Laminate_group_component_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_group_laminate_group_component_make_from_relationship is a type of Component_make_from_relationship (see 4.2.126). A Laminate_group_laminate_group_component_make_from_relationship is the association between one Assembly_group_component (see 4.2.40) and another Assembly_group_component, where one Assembly_group_component from full or partial reuse of the other. A Laminate_group_laminate_group_component_make_from_relationship also supports mapping of a Stratum in the reference design to a Stratum in this design.

NOTE The main use of this concept is use of multi-layer breakout definitions for complex integrated circuits where there is a mapping from a Stratum in the reference_footprint.reference_layout to a stratum in this layout.

The data associated with a Laminate_group_component_make_from_relationship are the following:

- resultant_component;
- reusable_component.

4.2.487.1 resultant_component

The resultant_component specifies the Assembly_group_component (see 4.2.40) that results from the reuse in the Laminate_group_laminate_group_component_make_from_relationship.

4.2.487.2 reusable_component

The reusable_component specifies the Assembly_group_component (see 4.2.40) that is reused to create the resultant_component in the Laminate_group_laminate_group_component_make_from_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY laminate_group_component_make_from_relationship
  SUBTYPE OF (component_make_from_relationship);
    SELF\component_make_from_relationship.reusable_component : assembly_group_component;
    SELF\component_make_from_relationship.resultant_component : assembly_group_component;
    stratum_mapping : SET [1:?] OF stratum_make_from_relationship;
WHERE
  WR1: SIZEOF(QUERY( sar &#60;* SELF\component_make_from_relationship.reusable_component.
    NOT('AP210_ARM.LAMINATE_COMPONENT' IN TYPEOF(sar.related_component))
    )) = 0;
  WR2: SIZEOF(QUERY( sar &#60;* SELF\component_make_from_relationship.resultant_component
    NOT('AP210_ARM.LAMINATE_COMPONENT' IN TYPEOF(sar.related_component))
    )) = 0;
END_ENTITY; -- laminate_group_component_make_from_relationship

( *

```

4.2.488 Laminate_text_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Laminate_text_component` is a type of `Laminate_component` (see 4.2.483) that is designed to be interpreted as a single character portion of a text message by a human or by an artificial vision system. A `Laminate_text_component` may be used to help compose a `Laminate_text_string_component` (see 4.2.489) to form a complete message. The composition will be defined by an instance of `Assembly_group_component` (see 4.2.40) that relates all necessary members of `Laminate_component` to compose the message embodied in the instance of `Laminate_text_string_component`. Each `Stratum_feature` referenced by the `stratum_feature_implementation` shall be on the same `Stratum` (see 4.2.845).

EXAMPLE 1 Figure 55 illustrates possible arrangements and orientation of the `Laminate_text_string_component` in the material of the `Stratum` (see 4.2.845). The figure includes multiple `Laminate_text_component` members presenting characters.

The data associated with a `Laminate_text_component` are the following:

- `of_character`;
- `stratum_feature_implementation`;
- `usage_definition`.

4.2.488.1 of_character

The `of_character` specifies the message content, a character, for the `Laminate_text_component`.

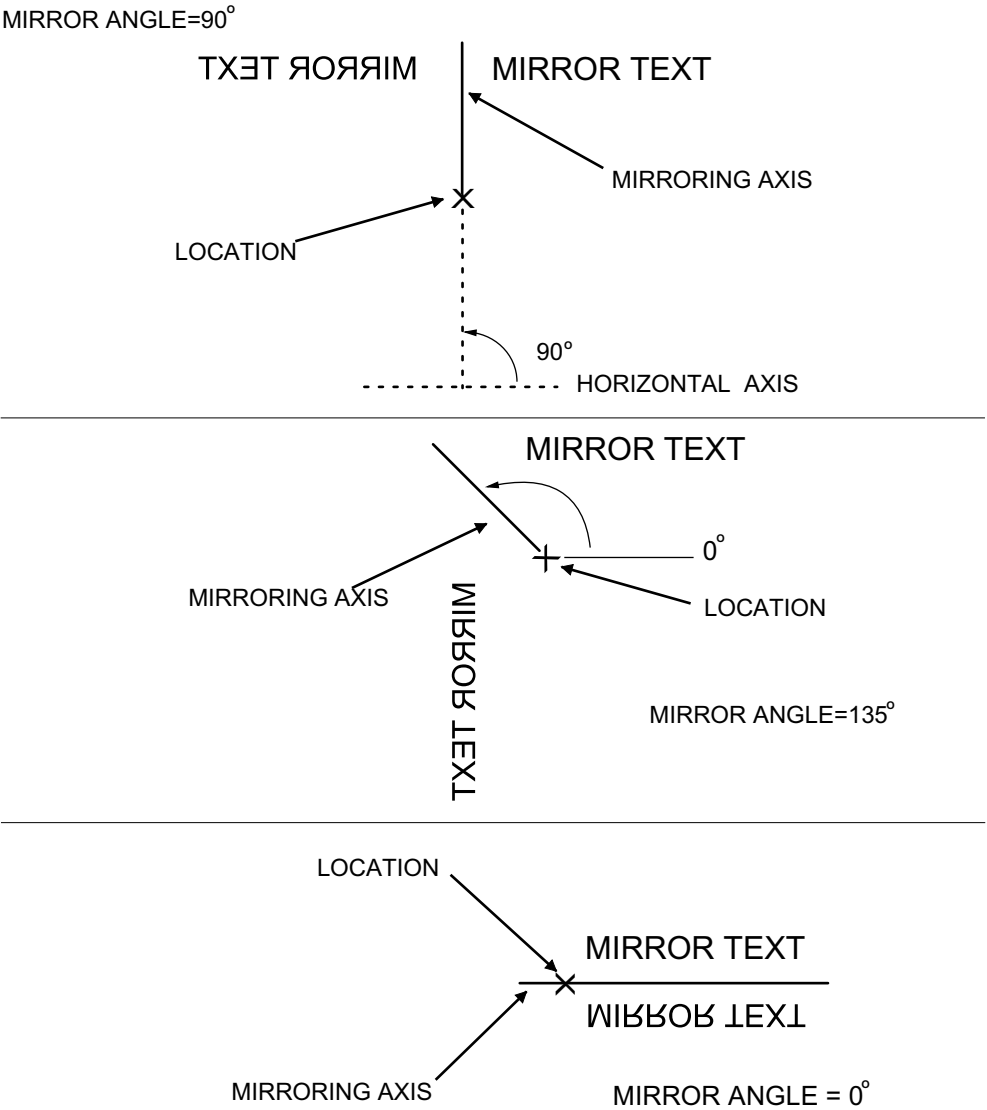


Figure 55 – Laminate_text_component

4.2.488.2 stratum_feature_implementation

The stratum_feature_implementation specifies the role of the Stratum_feature (see 4.2.851) for the Laminate_text_component. There may be more than one Stratum_feature for the Laminate_text_component.

NOTE Multiple Stratum_feature may exist in the case that the shape of the character is disjoint, but they are all on the same Stratum.

4.2.488.3 usage_definition

The usage_definition specifies the Part_text_template (see 4.2.632) that contains the font definition and associated parameters to be used for the Laminate_text_component.

NOTE EXPRESS specification:

```

*)
ENTITY laminate_text_component
  SUBTYPE OF (laminate_component);
    of_character                      : STRING (1) FIXED;
    SELF\assembly_component.usage_definition : part_text_template;
    stratum_feature_implementation      : SET [1:?] OF stratum_feature;
END_ENTITY; -- laminate_text_component

( *
```

4.2.489 Laminate_text_string_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Laminate_text_string_component is a type of Laminate_component (see 4.2.483) and is a type of Array_placement_group (see 4.2.29). A Laminate_text_string_component conveys a complete one line message to the viewer. A Laminate_text_string_component is composed by associating one or more individual instances of Laminate_text_component (see 4.2.488) to the Laminate_text_string_component using one instance of Sub_assembly_relationship (see 4.2.877). The Laminate_text_string_component itself does not directly refer to a font definition. The usage_definition inherited from the supertype Assembly_component (see 4.2.34) is unnecessary since this Application Object is defined by the composition of Laminate_text_component contributing to it. The position of the Laminate_text_string_component shall be interpreted as a constraint on the position of the individual components contributing to it. The position of the Laminate_text_string_component shall be interpreted as the insertion point for the message. The Laminate_text_string_component may have an orientation with respect to the design coordinate system, in which case the components composing the Laminate_text_string_component shall have a corresponding orientation. The Axis_placement (see 4.2.56) or Cartesian_transformation_operator_2d (see 4.2.88) which locates the Laminate_text_string_component shall be interpreted as a local coordinate system in order that any orientation shall be applied consistently to all components. An implementation shall ensure that adjacent characters that use different fonts still meet the requirement that Stratum_feature (see 4.2.851) instances are disjoint.

NOTE If multi-line messages are required, multiple instances of `Laminate_text_string_component` will be needed, one for each line of the message.

The data associated with a `Laminate_text_string_component` are the following:

- `message`;
- `of_horizontal_justification`;
- `of_vertical_justification`.

4.2.489.1 message

The message specifies the message content, a string, for the `Laminate_text_string_component`.

4.2.489.2 of_horizontal_justification

The `horizontal_justification` specifies either "left", "right", or "centre" for the beginning point of the message with respect to the position of the `Axis_placement` (see 4.2.56) or `Cartesian_transformation_operator_2d` (see 4.2.88) that is the insertion point for the text. The "left" indicates that the left edge of the bounding box of the first character of the message is aligned with the 'Y' axis of the `Axis_placement`. The "right" indicates that the right edge of the bounding box of the last character of the message is aligned with the 'Y' axis of the `Axis_placement`. The "centre" indicates that the left edge of the bounding box of the first character of the message and the right edge of the bounding box of the last character of the message are equally displaced on either side of the 'Y' axis of the `Axis_placement`.

4.2.489.3 of_vertical_justification

The `vertical_justification` specifies either "top", "bottom", or "centre" for the beginning point of the message with respect to the position of the `Axis_placement` (see 4.2.56) or `Cartesian_transformation_operator_2d` (see 4.2.88) that is the insertion point for the text. The "top" indicates that the top edge of the bounding box of the first character of the message is aligned with the 'X' axis of the `Axis_placement`. The "bottom" indicates that the bottom edge of the bounding box of the last character of the message is aligned with the 'X' axis of the `Axis_placement`. The "centre" indicates that the top edge of the bounding box of the first character of the message and the bottom edge of the bounding box of the last character of the message are equally displaced on either side of the 'X' axis of the `Axis_placement`.

NOTE EXPRESS specification:

```

*)
ENTITY laminate_text_string_component
  SUBTYPE OF (array_placement_group, laminate_component);
  message : message_content;
  of_horizontal_justification : horizontal_justification;
  of_vertical_justification : vertical_justification;
END_ENTITY; -- laminate_text_string_component

( *

```

4.2.490 Land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Land is a type of Stratum_feature_template_component (see 4.2.862). A Land may be a Component_termination_passage_and_contact_size_dependent_land (see 4.2.144), a Component_termination_passage_dependent_land (see 4.2.146), a Contact_size_dependent_land (see 4.2.194), a Via_dependent_land (see 4.2.950), or a Via_and_contact_size_dependent_land (see 4.2.948). A Land helps compose a Conductor or a Conductive_filled_area.

NOTE In the context of this standard, a Land may be external or internal to the interconnect.

EXAMPLE 1 Figure 56 illustrates the relationship of a Land to the other elements of a construction known in the domain as a padstack.

The data associated with a Land are the following:

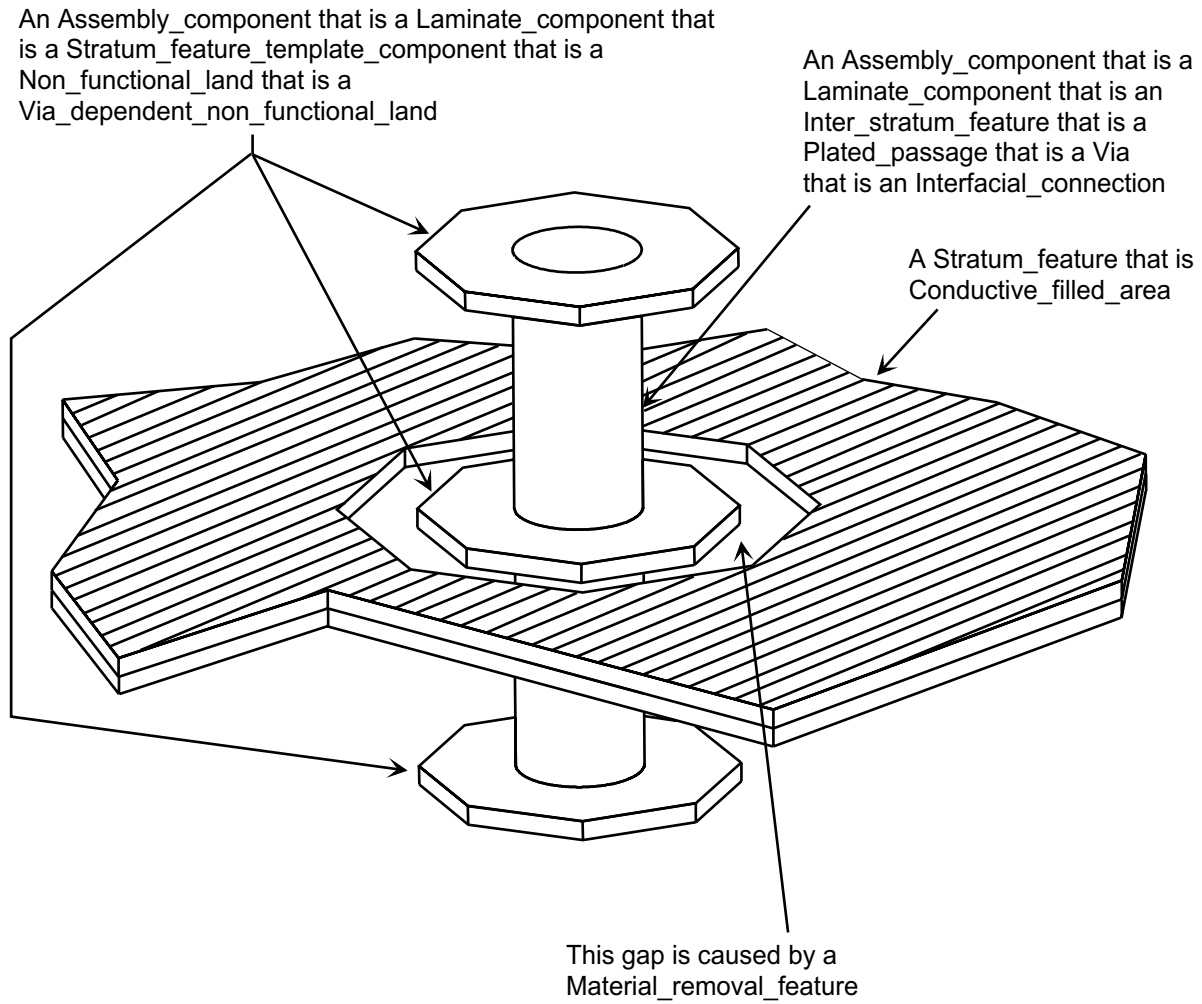
- access_mechanism;
- alternate_land_definition;
- usage_definition.

4.2.490.1 access_mechanism

The access_mechanism is an inverse relationship that specifies that the existence of the Land is dependent on the existence of the Land_join_terminal (see 4.2.492) that specifies the Land as its associated_component. There shall be one or more Land_join_terminal for a Land. This relationship is for internal access only.

4.2.490.2 alternate_land_definition

The alternate_land_definition specifies one role of the Land_physical_template (see 4.2.493) for the Land. The alternate_land_definition need not be specified for a particular Land.

**Figure 56 – Land**

4.2.490.3 usage_definition

The usage_definition specifies the primary role of the Land_physical_template (see 4.2.493) for the Land. The usage_definition need not be specified for a particular Land.

NOTE The Land_join_terminal allows an application to discover the usage_definition for the Land through the Land_join_terminal.definition.associated_definition path.

Constraints:

WR1: A member of land shall be directly connected to a member of conductive_interconnect_element or to a member of conductive_area.

WR2: When an alternate_land_definition is present it overrides the usage_definition for the Land.

NOTE EXPRESS specification:

```

*)
ENTITY land
  SUPERTYPE OF (ONEOF (via_dependent_land,
    component_termination_passage_dependent_land,
    contact_size_dependent_land,
    component_termination_passage_and_contact_size_dependent_land,
    via_and_contact_size_dependent_land))
  SUBTYPE OF (stratum_feature_template_component);
  alternate_land_definition          : OPTIONAL land_physical_template;
  SELF\assembly_component.usage_definition : OPTIONAL land_physical_template;
  INVERSE
    access_mechanism                : SET [1:?] OF
                                      land_join_terminal FOR associated_component;
END_ENTITY; -- land

( *
```

4.2.491 Land_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Land_interface_terminal is a type of Component_feature (see 4.2.118) included to identify the interface available to be fastened to in the next level of assembly.

The data associated with a Land_interface_terminal are the following:

- associated_component;
- definition.

4.2.491.1 associated_component

The associated_component specifies the role of the Land (see 4.2.490) for the Land_interface_terminal. Only the following types of Land shall be referenced: Contact_size_dependent_land; Component_termination_passage_and_contact_size_dependent_land; Via_and_contact_size_dependent_land.

4.2.491.2 definition

The definition specifies the role of the Land_template_interface_terminal (see 4.2.494) for the Land_interface_terminal.

NOTE Use of this capability occurs when a member of Component_feature_to_physical_usage_view_assignment references a member of Land_interface_terminal to define a traceability relationship to an Interconnect_module_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY land_interface_terminal
  SUBTYPE OF (laminar_component_interface_terminal);
    SELF\component_feature.definition      : land_template_interface_terminal;
    SELF\component_feature.associated_component : land;
WHERE
  WR1: SIZEOF([ 'AP210_ARM.CONTACT_SIZE_DEPENDENT_LAND',
                'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_LAND',
                'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND' ] *
              TYPEOF(SELF\component_feature.associated_component))
    = 1;
END_ENTITY; -- land_interface_terminal

( *
```

4.2.492 Land_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Land_join_terminal is a type of Component_terminal (see 4.2.137) included in the design to mate directly with a terminal of another fabricated feature by use of a Fabrication_joint (see 4.2.368) internally in a layout.

The data associated with a Land_join_terminal are the following:

- associated_component;
- definition.

4.2.492.1 associated_component

The associated_component specifies the role of the Land (see 4.2.490) for the Land_join_terminal.

4.2.492.2 definition

The definition specifies the role of the Land_template_join_terminal (see 4.2.495) for the Land_join_terminal.

Constraints:

WR1: A Land_join_terminal shall participate in a relationship with a Fabrication_joint as a fabrication_features.

NOTE EXPRESS specification:

```
*)
ENTITY land_join_terminal
  SUBTYPE OF (laminate_component_join_terminal);
    SELF\component_feature.definition      : land_template_join_terminal;
    SELF\component_feature.associated_component : land;
END_ENTITY; -- land_join_terminal
```

(*

4.2.493 Land_physical_template

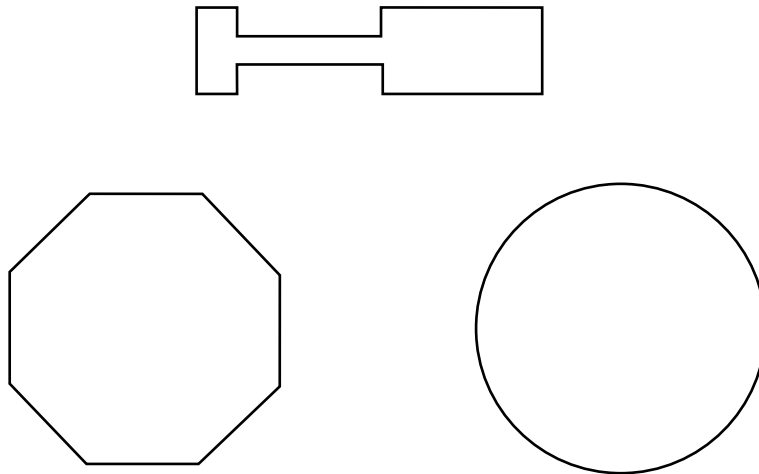
NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Land_physical_template is a type of Stratum_feature_template (see 4.2.861). A Land_physical_template may be a Default_land_physical_template (see 4.2.250). If the material the Land_physical_template is composed of is conductive, the Land_physical_template provides the functionality of a short among its terminals.

EXAMPLE 1 Figure 57 illustrates explicit geometry that represents simplified Land_physical_template.

The data associated with a Land_physical_template are the following:

- access_mechanisms;
- pre_defined_classification_code.



typical land shapes as defined in land_physical_template

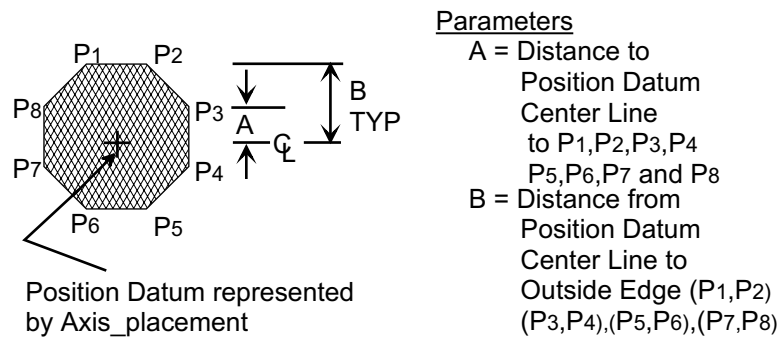
Figure 57 – Custom designs for Land_physical_template

4.2.493.1 access_mechanisms

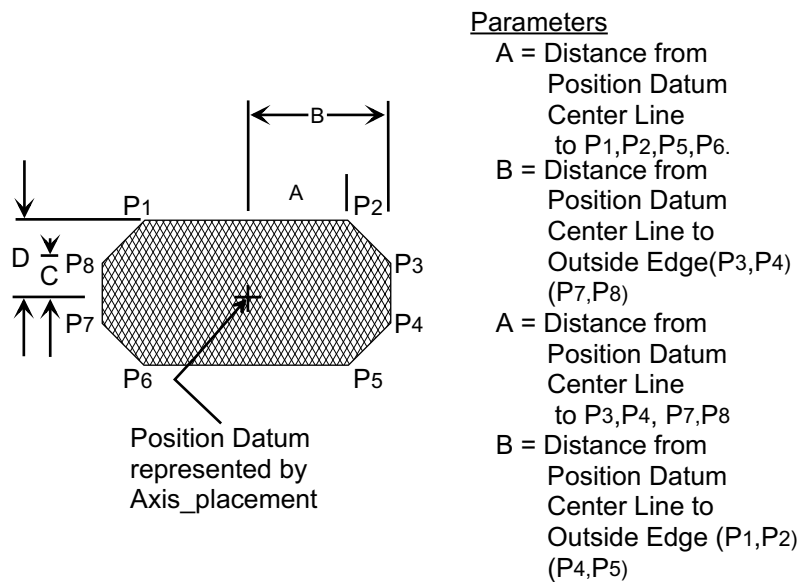
The access_mechanisms is an inverse relationship that specifies that the existence of the Land_physical_template is dependent on the existence of the Land_template_join_terminal (see 4.2.495) that specifies the Land_physical_template as its associated_definition. There shall be one or more Land_template_join_terminal for a Land_physical_template. Fabrication and assembly tolerances define the required shapes and variances for the terminal shapes.

4.2.493.2 pre_defined_classification_code

The pre_defined_classification_code specifies the role of the External_definition (see 4.2.361) for the Land_physical_template. The pre_defined_classification_code need not be specified for a specific Land_physical_template.



Variable Octagon



Variable Elongated Octagon

Physical characteristics are inherited from Part_template.
Reference document is : IEC 61182-1

Figure 58 – Complex land template shapes

NOTE 1 IEC 61182-1 provides a set of Land (see 4.2.490) identifiers. See figure 58.

NOTE 2 The Land_template, Usage_concept_usage_relationship (see 4.2.942), and the Geometric Dimensioning and Tolerancing capabilities of this part of ISO 10303 may be used to unambiguously exchange the shape of both pre-defined Land patterns that include only geometry (as from IEC 61182-1) and enterprise defined patterns. When this facility is used to create more than one equivalent Land definition, it may not be possible to maintain knowledge of the connection functionality of the Land definition. This is because Land is an atomic concept in this part of ISO 10303.

NOTE 3 The Assembly_group_component (see 4.2.40), Interconnect_module (see 4.2.440), and the Geometric Dimensioning and Tolerancing capabilities of this part of ISO 10303 may be used to unambiguously exchange both the shape and connection knowledge of Land patterns that include pre-defined geometry (as from IEC 61182-1) and enterprise defined geometry that have explicit definitions of the fabrication and assembly constraints.

NOTE 4 The Land_physical_template inherit material property data from Part_template.

Constraints:

UR1: The pre_defined_classification_code shall be unique within a population of Land_physical_templates.

NOTE EXPRESS specification:

```

*)
ENTITY land_physical_template
  SUPERTYPE OF (default_land_physical_template)
  SUBTYPE OF (stratum_feature_template);
  pre_defined_classification_code : OPTIONAL external_definition;
  INVERSE
    access_mechanisms : SET [1:?] OF land_template_terminal FOR
      associated_definition;
  UNIQUE
    UR1 : pre_defined_classification_code;
WHERE
  WR1: SIZEOF(QUERY(ltjt &#60;* access_mechanisms |
    NOT('AP210_ARM.LAND_TEMPLATE_JOIN_TERMINAL' IN TYPEOF(ltjt))
  )) = 0;
END_ENTITY; -- land_physical_template

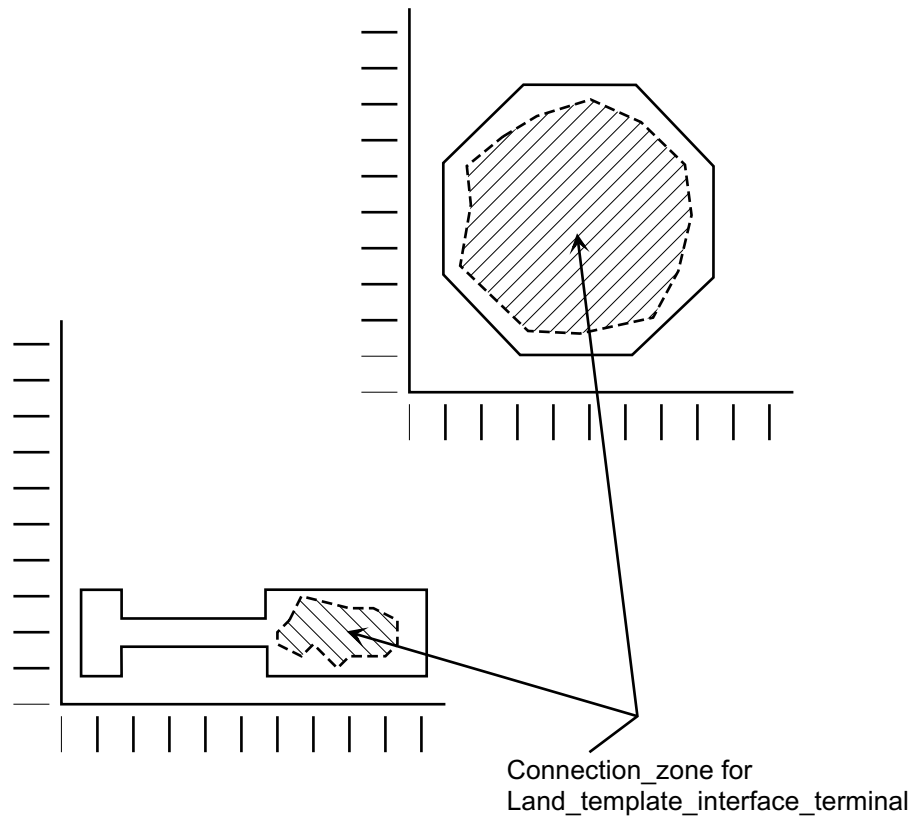
( *
```

4.2.494 Land_template_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Land_template_interface_terminal` is a type of `Land_template_terminal` (see 4.2.496) that has an external access functionality. A `Land_template_interface_terminal` redefines the `associated_definition` attribute inherited from `Land_template_terminal` to only reference a `Default_attachment_size_based_land_physical_template` (see 4.2.247).

EXAMPLE 1 Figure 59 is a plan view that illustrates the geometry that represents the interface terminal for a `Land_physical_template` (see 4.2.493).



Note : The usage view associated with this data is the `Land_template`

Figure 59 – `Land_template_interface_terminal`

NOTE External refers to something that is outside of the product that an occurrence is helping to compose.

NOTE EXPRESS specification:

```

*)
ENTITY land_template_interface_terminal
  SUBTYPE OF (land_template_terminal);
  SELF\land_template_terminal.associated_definition :
    default_attachment_size_based_land_physical_template;
END_ENTITY; -- land_template_interface_terminal

( *

```

4.2.495 Land_template_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Land_template_join_terminal is a type of Land_template_terminal (see 4.2.496) that has an internal access functionality.

EXAMPLE 1 Figure 60 is a plan view that illustrates the geometry that represents the join terminal for a Land_physical_template (see 4.2.493). In the plan view, the join terminal is on the periphery.

NOTE Internal refers to something that is intrinsic to the product of which an occurrence is part.

NOTE EXPRESS specification:

```

*)
ENTITY land_template_join_terminal
  SUBTYPE OF (land_template_terminal);
END_ENTITY; -- land_template_join_terminal

( *

```

4.2.496 Land_template_terminal

A Land_template_terminal is a type of Shape_element (see 4.2.810). A Land_template_terminal is either a Land_template_interface_terminal (see 4.2.494) or a Land_template_join_terminal (see 4.2.495). The connection_zone_category shall be supplied if the connection_zone is supplied. Each member of connection_zone_category shall correspond to exactly one member of terminal_connection_zone.

NOTE The binding of a member of connection_zone_category to the member of terminal_connection_zone is formally defined in the mapping table in clause 5.1

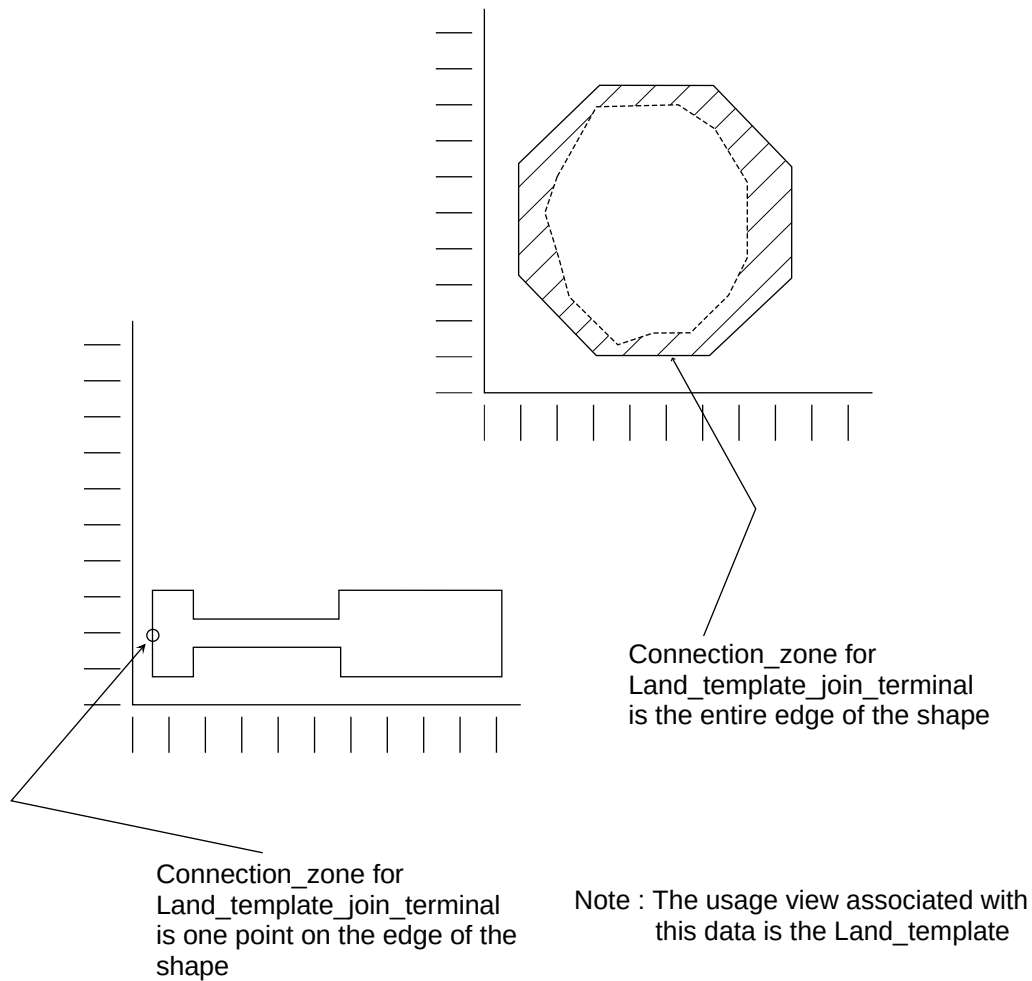


Figure 60 – Land_template_join_terminal

The data associated with a Land_template_terminal are the following:

- associated_definition;
- connection_zone_category;
- name;
- terminal_connection_zone.

4.2.496.1 associated_definition

The associated_definition specifies the role of the Land_physical_template (see 4.2.493) for the Land_template_terminal.

4.2.496.2 connection_zone_category

The connection_zone_category specifies either an edge_curve, an edge_point, a surface_area, or a surface_point as a category of the Connection_zone (see 4.2.185) for the Land_template_terminal. The connection_zone_category is optional and need not be supplied. There may be zero or more connection_zone_category. The interpretation of this attribute shall be to the description of the relationship between the terminal and each individual connection zone.

4.2.496.3 name

The name identifies the Land_template_terminal.

4.2.496.4 terminal_connection_zone

The terminal_connection_zone specifies the role of the Connection_zone (see 4.2.185) for the Land_template_terminal. There may be one or more Connection_zone for the Land_template_terminal. The terminal_connection_zone is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the terminal_connection_zone shall be supplied.

Constraints:

UR1: The combination of the associated_definition and the name shall be unique within a population of Land_template_terminals.

WR1: The points comprising the terminal_connection_zone must reside in the shape of the electrical device being accessed.

WR2: The points comprising the terminal_connection_zone shall be defined in the coordinate system of the device being accessed.

WR3: The connection_zone_category shall be provided if the terminal_connection_zone is provided.

WR4: The size of connection_zone_category shall be equal to the sizeof terminal_connection_zone if the terminal_connection_zone is provided.

WR5: Each member of connection_zone_category shall be for exactly one member of terminal_connection_zone.

NOTE EXPRESS specification:

```

*)
ENTITY land_template_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (land_template_join_terminal,
    land_template_interface_terminal))
  SUBTYPE OF (shape_element);
    associated_definition      : land_physical_template;
    terminal_connection_zone   : OPTIONAL SET [1:?] OF connection_zone;
    SELF\shape_element.name    : ee_name;
    connection_zone_category   : OPTIONAL SET [1:?] OF land_template_terminal_class;
INVERSE
  terminal_shape              : SET [0:1] OF usage_concept_usage_relationship FOR
    associated_usage;
  UNIQUE
    UR1 : name, associated_definition;
  WHERE
    WR1 : NOT(EXISTS(terminal_connection_zone)) OR
      EXISTS(connection_zone_category);
    WR2 : NOT(EXISTS(terminal_connection_zone)) OR
      (SIZEOF(terminal_connection_zone) =
        SIZEOF(connection_zone_category));
END_ENTITY; -- land_template_terminal
( *

```

4.2.497 Language_reference_manual

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Language_reference_manual is a type of Ee_specification (see 4.2.338) that defines the syntax and semantics of a computer interpretable language.

NOTE EXPRESS specification:

```

*)
ENTITY language_reference_manual
  SUBTYPE OF (ee_specification);
END_ENTITY; -- language_reference_manual

( *

```

4.2.498 Layer

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layer is a type of `Managed_design_object` (see 4.2.524). A Layer is an extent of contiguous `Stratum` (see 4.2.845) in a printed circuit board. The `Stratum_feature` (see 4.2.851) of all `Stratum` of a Layer are as alike in shape as process tolerances will allow; the design intent is that the shapes of the `Stratum_features` are to be identical. Standard usage is for exactly one of a `Design_layer_stratum` (see 4.2.275) or `Documentation_layer_stratum` (see 4.2.306) to be included in the `Inter_stratum_extent` (see 4.2.431) referenced by extent.

The data associated with a Layer are the following:

- extent;
- layer_usage;
- name.

4.2.498.1 extent

The extent specifies the role of the `Inter_stratum_extent` (see 4.2.431) for the Layer.

4.2.498.2 layer_usage

The `layer_usage` specifies the `External_definition` (see 4.2.361) that characterizes the purpose for incorporating the Layer into a `Interconnect_module` (see 4.2.440) design. The `layer_usage` need not be specified for a particular Layer.

4.2.498.3 name

The name identifies the Layer within the product.

Constraints:

UR1: The name shall be unique within a population of Layers for a product.

WR1: Each member of `Stratum` specified by Layer shall be represented by the same geometric representation.

WR2: Each member of Layer shall be related to no more than one `Design_layer_stratum` or no more than one `Documentation_layer_stratum`.

NOTE EXPRESS specification:

```

*)
ENTITY layer
  SUBTYPE OF (managed_design_object);
  extent      : inter_stratum_extent;
  layer_usage : OPTIONAL external_definition;
  name        : ee_name;
  UNIQUE
    UR1 : name;
END_ENTITY; -- layer

--tt tc1 wd10 modified
--tt tc1 wd9 modified
--tt tc1 wd8 modified
( *
```

4.2.499 Layer_connection_point

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layer_connection_point is a type of Managed_design_object (see 4.2.524). A Layer_connection_point is either a Dependently_located_layer_connection_point (see 4.2.259) or an Explicitly_located_layer_connection_point (see 4.2.360) that resides on the surface of a Design_layer_stratum (see 4.2.275) in the Interconnect_module (see 4.2.440) and is defined in the coordinate system of the Interconnect_module. The Layer_connection_point Application Object is an intersection concept between the topology requirements, placement requirements defined in a geometric coordinate system and the Stratum material definition.

NOTE A Layer_connection_point may be placed in an area where the conductive material is removed since there is no information contained in this Application Object regarding vertical material processes. It is the abstraction in this standard to separate lateral material description from the vertical material modification processes.

The data associated with a Layer_connection_point are the following:

- join_path;
- name;
- resident_design_layer_stratum.

4.2.499.1 join_path

The join_path is an inverse relationship that specifies that the existence of the Layer_connection_point is dependent on the existence of the Join_relationship (see 4.2.479) that specifies the Layer_connection_point as its points_to_be_connected. There shall be one or more Join_relationship for a Layer_connection_point.

4.2.499.2 name

The name identifies a Layer_connection_point.

4.2.499.3 resident_design_layer_stratum

The resident_design_layer_stratum specifies the Design_layer_stratum (see 4.2.275) upon which the Layer_connection_point is located.

Constraints:

UR1: The name, in combination with the resident_design_layer_stratum, shall be unique within a population of Layer_connection_point.

WR1: For each member of Cartesian_point associated with a Design_layer_stratum, there shall be no more than one Layer_connection_point associated with that Cartesian_point.

WR2: Each member of Layer_connection_point shall be specified by no more than one member of Physical_network.

NOTE EXPRESS specification:

```

*)
ENTITY layer_connection_point
  ABSTRACT SUPERTYPE OF (ONEOF (
    explicitly_located_layer_connection_point,
    dependently_located_layer_connection_point))
  SUBTYPE OF (managed_design_object);
  location_2d          : OPTIONAL cartesian_point;
  location_3d          : OPTIONAL cartesian_point;
  name                 : ee_name;
  resident_design_layer_stratum : design_layer_stratum;
DERIVE
  referencing_network : SET OF physical_network :=
    QUERY(ijp &#60;* inter_join_path | SIZEOF(USEDIN(ijp,
    'AP210_ARM.PHYSICAL_NETWORK.' +
    'INTERCONNECT_MODULE_NETWORK_TOPOLOGY_ELEMENT') +
    USEDIN(intra_join_path,
    'AP210_ARM.PHYSICAL_NETWORK.' +
    'INTERCONNECT_MODULE_NETWORK_TOPOLOGY_ELEMENT')) &#62; 0);
INVERSE
  inter_join_path      : SET [0:1] OF inter_stratum_join_relationship FOR
    points_to_be_connected;
  intra_join_path      : intra_stratum_join_relationship FOR
    points_to_be_connected;
UNIQUE
  UR1 : name, resident_design_layer_stratum;
WHERE
  WR1: EXISTS(location_2d) OR EXISTS(location_3d);
  WR2: NOT(EXISTS(location_2d)) OR
    (location_2d\shape_item.
    geometric_context.dimensionality = two_dimensional);
  WR3: NOT(EXISTS(location_2d)) OR
    (location_3d\shape_item.
    geometric_context.dimensionality = three_dimensional);
  WR4: SIZEOF(referencing_network) &#60;= 1;
END_ENTITY; -- layer_connection_point
(

```

4.2.500 Layer_connection_point_link

A Layer_connection_point_link is a type of Managed_design_object (see 4.2.524). A Layer_connection_point_link is a directed link between two Layer_connection_point (see 4.2.499) in order to support the sequence of Layer_connection_point required in Unrouted_conductive_interconnect_element (see 4.2.937).

The data associated with a `Layer_connection_point_link` are the following:

- `associated_stratum`;
- `precedent_point`;
- `subsequent_point`.

4.2.500.1 associated_stratum

An `associated_stratum` specifies the role of the `Stratum` (see 4.2.845) for a `Layer_connection_point_link`.

4.2.500.2 precedent_point

A `precedent_point` specifies one role of the `Layer_connection_point` (see 4.2.499) for a `Layer_connection_point_link`.

4.2.500.3 subsequent_point

A `subsequent_point` specifies one role of the `Layer_connection_point` (see 4.2.499) for a `Layer_connection_point_link`.

Constraints:

- WR1:** The `precedent_point` shall not be the `subsequent_point`.
- WR2:** The `precedent_point` shall be on the `associated_stratum`.
- WR3:** The `subsequent_point` shall be on the `associated_stratum`.

NOTE EXPRESS specification:

```

*)
ENTITY layer_connection_point_link
  SUBTYPE OF (managed_design_object);
    associated_stratum : stratum;
    precedent_point    : layer_connection_point;
    subsequent_point   : layer_connection_point;
INVERSE
  referenced_by : unrouted_conductive_interconnect_element FOR
    connected_points;
WHERE
  WR1 : precedent_point &#60;&#62; subsequent_point;
  WR2 : associated_stratum = precedent_point.resident_design_layer_stratum;
  WR3 : associated_stratum = subsequent_point.resident_design_layer_stratum;
END_ENTITY; -- layer_connection_point_link

( *

```

4.2.501 Layer_qualified_layout_spacing_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layer_qualified_layout_spacing_requirement_occurrence is a type of Layout_spacing_requirement_occurrence (see 4.2.509) that describes the physical spacing between one of an interconnect_module_cutout, an interconnect_module_fill_area, an interconnect_module_inter_stratum_feature, an interconnect_module_stratum_feature, or an interconnect_module_via, and another of these categories of design objects. This Application object qualifies the spacing requirement to apply within a given context as defined by a set of Stratum (see 4.2.845).

The category names correspond to the category of equivalent application object names as described in Layout_spacing_requirement_occurrence. This requirement shall be interpreted as applying to the categories of those objects.

NOTE A pair of categories is used for this requirement occurrence.

EXAMPLE 1 In domain terms, when a Via to Via spacing is to be greater than 5 mm, the encoding will be that the spacing between the reference_design_object_category whose value is interconnect_module_via and the dependent_design_object_category whose value is interconnect_module_via shall be greater than 5 mm.

EXAMPLE 2 In domain terms, when the distance from a cutout to a trace must be greater than 5 mm, the encoding will be that the spacing between the reference_design_object_category whose value is interconnect_module_cutout and the dependent_design_object_category whose value is interconnect_module_stratum_feature shall be greater than 5 mm.

The data associated with a Layer_qualified_layout_spacing_requirement_occurrence are the following:

— layer_context.

4.2.501.1 layer_context

The layer_context specifies an Inter_stratum_extent (see 4.2.431), for the Layer_qualified_layout_spacing_requirement_occurrence. The layer_context establishes a context within which the spacing requirement is to be met. There may be more than one Inter_stratum_extent for a Layer_qualified_layout_spacing_requirement_occurrence. The Inter_stratum_extent may overlap. All Inter_stratum_extent specified shall be in the same design.

NOTE EXPRESS specification:

```
*)
ENTITY layer_qualified_layout_spacing_requirement_occurrence
  SUBTYPE OF (layout_spacing_requirement_occurrence);
    layer_context : SET [1:?] OF inter_stratum_extent;
END_ENTITY; -- layer_qualified_layout_spacing_requirement_occurrence

( *
```

4.2.502 Layout_land_width_tolerance_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_land_width_tolerance_requirement_occurrence is a type of Inspection_requirement_occurrence (see 4.2.430) that characterizes the land width range that is required to be produced for a particular terminal pitch class of mating components.

The data associated with a Layout_land_width_tolerance_requirement_occurrence are the following:

- mating_feature_pitch_class;
- maximum_negative_deviation.

4.2.502.1 mating_feature_pitch_class

The mating_feature_pitch_class specifies the role of the Pitch_class (see 4.2.673) for the Layout_land_width_tolerance_requirement_occurrence.

4.2.502.2 maximum_negative_deviation

The maximum_negative_deviation specifies the largest negative Length_data_element (see 4.2.511) allowed width deviation for the Layout_land_width_tolerance_requirement_occurrence. The deviation is measured at the surface of the Land that mates with the external device.

NOTE The measurement location deviates from normal fabrication practice of measuring at the base of the Land in order to guarantee successful assembly process.

NOTE EXPRESS specification:

```

*)
ENTITY layout_land_width_tolerance_requirement_occurrence
  SUBTYPE OF (inspection_requirement_occurrence);
    mating_feature_pitch_class      : pitch_class;
    maximum_negative_deviation      : length_data_element;
END_ENTITY; -- layout_land_width_tolerance_requirement_occurrence

( *

```

4.2.503 Layout_line_width_tolerance_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_line_width_tolerance_requirement_occurrence is a type of Inspection_requirement_occurrence (see 4.2.430) that characterizes the line width range that is expected by the design organization to be manufactured for a particular Stratum_technology.

The data associated with a Layout_line_width_tolerance_requirement_occurrence are the following:

- characterized_stratum_technology;
- maximum_negative_deviation;
- maximum_positive_deviation;
- tolerance_line_width_class.

4.2.503.1 characterized_stratum_technology

The characterized_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Layout_line_width_tolerance_requirement_occurrence.

4.2.503.2 maximum_negative_deviation

The maximum_negative_deviation specifies the largest negative Length_data_element (see 4.2.511) for the Layout_line_width_tolerance_requirement_occurrence.

4.2.503.3 maximum_positive_deviation

The maximum_positive_deviation specifies the largest positive Length_data_element (see 4.2.511) for the Layout_line_width_tolerance_requirement_occurrence.

4.2.503.4 tolerance_line_width_class

The tolerance_line_width_class specifies the role of the Line_width_class (see 4.2.515) for the Layout_line_width_tolerance_requirement_occurrence.

NOTE EXPRESS specification:

```

*)
ENTITY layout_line_width_tolerance_requirement_occurrence
  SUBTYPE OF (inspection_requirement_occurrence);
    characterized_stratum_technology : stratum_technology;
    tolerance_line_width_class       : line_width_class;
    maximum_negative_deviation       : length_data_element;
    maximum_positive_deviation       : length_data_element;
END_ENTITY; -- layout_line_width_tolerance_requirement_occurrence

( *
```

4.2.504 Layout_macro_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_macro_component is a type of Printed_component (see 4.2.721). A Layout_macro_component uses a Layout_macro_floor_plan_template (see 4.2.507) as the usage definition for the shape of the unexpanded macro in the design. The design definition is the contents of the macro to be expanded. A Layout_macro_component may be exchanged as part of a recursive layout definition. A Layout_macro_component itself generates no Stratum_features since it is not expanded. A Layout_macro_component does inherit terminals from the Printed_component. These terminals are used in the floor planning layout process. These terminals may be on any layer necessary in the design.

NOTE The recursive definition must end on a Laminate_component that is not a Layout_macro_component.

The data associated with a Layout_macro_component are the following:

- design_definition;
- usage_definition.

4.2.504.1 design_definition

The design_definition specifies the role of a Layout_macro_definition (see 4.2.505) for the Layout_macro_component.

4.2.504.2 usage_definition

The usage_definition specifies the role of a Layout_macro_floor_plan_template (see 4.2.507) for the Layout_macro_component.

NOTE EXPRESS specification:

```

*)
ENTITY layout_macro_component
  SUBTYPE OF (printed_component);
    design_definition : layout_macro_definition;
    SELF\printed_component.usage_definition : layout_macro_floor_plan_template;
WHERE
WR1: NOT(EXISTS (SELF\printed_component.stratum_feature_implementation))
    AND
    NOT(EXISTS (SELF\printed_component.printed_component_stack))
    AND
    NOT(EXISTS (SELF\printed_component.associated_precedent_link))
    AND
    NOT(EXISTS (SELF\printed_component.associated_subsequent_link))
    AND
    NOT(EXISTS (SELF\printed_component.required_material_stack));
END_ENTITY; -- layout_macro_component
( *
```

4.2.505 Layout_macro_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_macro_definition is a type of Pcb (see 4.2.642). A Layout_macro_definition is a design view of a interconnect structure that is intended to be reused as a structure rather than as a component. A Layout_macro_definition may be an aggregation of Stratum (see 4.2.845). A Layout_macro_definition may be an aggregation of Layout_macro_definition_component (see ??) A Layout_macro_definition shall not be used as a component in a physical assembly.

EXAMPLE 1 A Layout_macro_definition may be used as part of a recursive definition or subroutine layout.

The data associated with a Layout_macro_definition are the following:

— floor_layout_view.

4.2.505.1 floor_layout_view

The floor_layout_view specifies the Layout_macro_floor_plan_template (see 4.2.507) for the Layout_macro_definition as a layout template for the floor planning layout. This template shall provide the terminal definitions for the Layout_macro_definition and a shape for the floor planning layout to enable the layout design without instantiating the design details of the Layout_macro_definition itself.

NOTE The floor_layout_view is a replacement for the usage_view that would be inherited from the Interconnect_module.

Constraints:

WR1: A Layout_macro_definition shall not be referenced in the role of usage_definition.

WR2: Layout_macro_definition.usage_view (this is an inherited attribute from Physical_unit_design_view) shall not exist.

NOTE EXPRESS specification:

```

*)
ENTITY layout_macro_definition
  SUBTYPE OF (pcb);
  floor_layout_view : layout_macro_floor_plan_template;
WHERE
  WR1: NOT(EXISTS(SELF\physical_unit_design_view.usage_view));
  WR2: NOT(EXISTS(SELF\interconnect_module.assembly_design_requirement));
  WR3: SIZEOF(USEDIN(SELF,
    'AP210_ARM.ASSEMBLY_COMPONENT.USAGE_DEFINITION')) = 0;
END_ENTITY; -- layout_macro_definition

( *
```

4.2.506 Layout_macro_definition_terminal_to_usage_terminal_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_macro_definition_terminal_to_usage_terminal_assignment is a type of Managed_design_object (see 4.2.524). A Layout_macro_definition_terminal_to_usage_terminal_assignment is the association between the Printed_part_template_terminal (see 4.2.737) and one of several fabricated features. The purpose for this assignment is to link the product data in the macro definition to the floor planning usage view to support a recursive definition. The Layout_macro_definition_terminal_to_usage_terminal_assignment shall be referenced by one Layout_macro_floor_plan_template.

NOTE 1 In a recursive definition, both a macro definition and the macro interface declaration that is the Layout_macro_floor_plan_template are archived together, with this relationship populated as necessary to provide linkages between the fabricated features in the macro definition and the features in the planning view. Without this linkage populated, there is no way to traverse the recursive definition so as to successfully resolve the terminals at each level of recursion.

NOTE 2 This facility is used to support transport of CAD data where geometric subroutines that include layer data and terminal data are defined.

The data associated with a Layout_macro_definition_terminal_to_usage_terminal_assignment are the following:

- assigned_design_object;
- defined_template_feature.

4.2.506.1 assigned_design_object

The assigned_design_object specifies the role of the Component_termination_passage_join_terminal (see 4.2.149) or of the Via_terminal (see 4.2.954) or of the Printed_component_join_terminal (see 4.2.722) or of the Conductive_interconnect_element_terminal (see 4.2.172) or of the Land_join_terminal (see 4.2.492) or of the Non_functional_land_join_terminal (see 4.2.564) for the Layout_macro_definition_terminal_to_usage_terminal_assignment.

4.2.506.2 defined_template_feature

The defined_template_feature specifies the role of the Printed_part_template_terminal (see 4.2.737) for the Layout_macro_definition_terminal_to_usage_terminal_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY layout_macro_definition_terminal_to_usage_terminal_assignment
  SUBTYPE OF (managed_design_object);
    defined_template_feature : printed_part_template_terminal;
    assigned_design_object   : fabricated_feature_select;
WHERE
  WR1: SIZEOF(USEDIN(SELF,
    'AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE.ACCESS_MAPS')) = 1;
END_ENTITY; -- layout_macro_definition_terminal_to_usage_terminal_assignment

( *
```

4.2.507 Layout_macro_floor_plan_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Layout_macro_floor_plan_template` is a type of `Printed_part_template` (see 4.2.730). A `Layout_macro_floor_plan_template` is part of a recursive definition or a geometric subroutine. A `Layout_macro_floor_plan_template` provides a layout shape and a set of terminal definitions for floor planning and layout purposes. The application compiler will replace the layout shape provided by this application object with the detailed definition specified by the `Layout_macro_definition` referenced by the `macro_definition` attribute.

NOTE 1 An occurrence of this object is a `Layout_macro_component` (see 4.2.504) that is instantiated during floor planning layout during the design process for the interconnect substrate.

NOTE 2 The shape of the `Layout_macro_floor_plan_template` should encompass the shape of the macro definition.

The data associated with a `Layout_macro_floor_plan_template` are the following:

- `access_mechanisms`;
- `characterized_material`.

4.2.507.1 access_mechanisms

The `access_maps` specifies the `Layout_macro_definition_terminal_to_usage_terminal_assignment` that relates the `Printed_part_template_terminal` that is an `access_mechanisms` inherited from the `Printed_part_template` with the fabrication features in the detailed definition. There shall be one or more `Layout_macro_definition_terminal_to_usage_terminal_assignment` for a `Layout_macro_floor_plan_template`.

4.2.507.2 characterized_material

The `macro_definition` is an inverse relationship that specifies that the existence of the `Layout_macro_floor_plan_template` is dependent on the existence of a `Layout_macro_definition` that specifies the `Layout_macro_floor_plan_template` as a `floor_layout_view`. There shall be zero or one `Layout_macro_definition` for a `Layout_macro_floor_plan_template`.

NOTE EXPRESS specification:

```

*)
ENTITY layout_macro_floor_plan_template
  SUBTYPE OF (printed_part_template);
    access_maps : SET [1:?] OF
      layout_macro_definition_terminal_to_usage_terminal_assignment;
INVERSE
  macro_definition : SET [0:1] OF layout_macro_definition FOR
    floor_layout_view;
WHERE
WR1: NOT(EXISTS (SELF\printed_part_template.of_stratum_technology))
    AND
    NOT(EXISTS (SELF\printed_part_template.material_stack))
    AND
    NOT(EXISTS (SELF\printed_part_template.printed_part_template_stack));
END_ENTITY; -- layout_macro_floor_plan_template

( *
```

4.2.508 Layout_spacing_requirement_non_conforming_design_object

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Layout_spacing_requirement_non_conforming_design_object is a type of Managed_design_object (see 4.2.524). A Layout_spacing_requirement_non_conforming_design_object specifies one or more design object that does not meet object to object spacing requirements for the printed circuit board layout.

The data associated with a Layout_spacing_requirement_non_conforming_design_object are the following:

- design_requirement;
- non_conformant_design_object;
- reference_design_object.

4.2.508.1 design_requirement

The design_requirement specifies the role of the Layout_spacing_requirement_occurrence (see 4.2.509) for the Layout_spacing_requirement_non_conforming_design_object

4.2.508.2 non_conformant_design_object

The non_conformant_design_object specifies one role of either an Assembly_component (see 4.2.34), a Component_feature (see 4.2.118), or one of a Component_termination_passage (see 4.2.143), a Conductive_filled_area (see 4.2.170), a Conductor (see 4.2.176), a Cutout (see 4.2.208), an Inter_stratum_feature (see 4.2.432), a Laminate_component (see 4.2.483), a Land (see 4.2.490), a Non_functional_land (see 4.2.562), a Stratum_feature (see 4.2.851), or a Via (see 4.2.947) for the Layout_spacing_requirement_non_conforming_design_object. There shall be one or more Assembly_component, Component_feature, Component_termination_passage, Conductive_filled_area, Conductor, Cutout, Inter_stratum_feature, Laminate_component, Land, Non_functional_land, Stratum_feature, or Via for each Layout_spacing_requirement_non_conforming_design_object.

4.2.508.3 reference_design_object

The reference_design_object specifies one role of either an Assembly_component (see 4.2.34), a Component_feature (see 4.2.118), or one of a Component_termination_passage (see 4.2.143), a Conductive_filled_area (see 4.2.170), a Conductor (see 4.2.176), a Cutout (see 4.2.208), an Inter_stratum_feature (see 4.2.432), a Laminate_component (see 4.2.483), a Land (see 4.2.490), a Non_functional_land (see 4.2.562), a Stratum_feature (see 4.2.851), or a Via (see 4.2.947) for the Layout_spacing_requirement_non_conforming_design_object.

NOTE This data is the result of an evaluation of the difference between the board layout and the original spacing requirements.

NOTE EXPRESS specification:

```

*)
ENTITY layout_spacing_requirement_non_conforming_design_object
  SUBTYPE OF (managed_design_object);
  non_conformant_design_object :
SET [1:?] OF assembly_or_interconnect_design_object_select;
  reference_design_object      : assembly_or_interconnect_design_object_select;
  design_requirement           : layout_spacing_requirement_occurrence;
END_ENTITY; -- layout_spacing_requirement_non_conforming_design_object

( *
```

4.2.509 Layout_spacing_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Layout_spacing_requirement_occurrence` is a type of `Ee_requirement_occurrence` (see 4.2.330) that describes the physical spacing between one of an `assembly_module_assembly_component`, an `assembly_module_component_feature`, an `interconnect_module_cutout`, an `interconnect_module_fill_area`, an `interconnect_module_design_intent_modification`, an `interconnect_module_inter_stratum_feature`, an `interconnect_module_cutout_edge_segment`, an `interconnect_module_interconnect_module_edge`, an `interconnect_module_interconnect_module_edge_segment`, an `interconnect_module_plated_cutout`, an `interconnect_module_plated_cutout_edge_segment`, an `interconnect_module_plated_interconnect_module_edge`, an `interconnect_module_plated_interconnect_module_edge_segment`, an `interconnect_module_plated_passage`, a category consisting of the union of an `interconnect_module_plated_passage` category and an `interconnect_module_plated_unsupported_passage` category, an `interconnect_module_unsupported_passage`, an `interconnect_module_stratum_feature`, or an `interconnect_module_via`, and another of these categories of design objects. A `Layout_spacing_requirement_occurrence` may be a `Layer_qualified_layout_spacing_requirement_occurrence`. The requirement shall be imposed between the boundaries of members of the the relevant categories or between centroids.

The category names correspond to the category of equivalent application object names.

NOTE Support for requirements between boundaries and centroids is not supported by this part of ISO 10303.

EXAMPLE 1 `Assembly_module_assembly_component` corresponds to `Assembly_component` (see 4.2.34) category. `Assembly_module_component_feature` corresponds to `Component_feature` (see 4.2.118) category. `Interconnect_module_cutout` corresponds to `Cutout` (see 4.2.208) category. `Interconnect_module_fill_area` corresponds to `Fill_area` (see 4.2.376) category. `Interconnect_module_inter_stratum_feature` corresponds to `Inter_stratum_feature` (see 4.2.432) category. `Interconnect_module_stratum_feature` corresponds to `Stratum_feature` (see 4.2.851) category. `Interconnect_module_via` corresponds to `Via` (see 4.2.947) category.

Any subtype of an item in a category shall qualify as part of that category.

NOTE 1 This part of ISO 10303 supports several spacing requirements in this Application Object.

NOTE 2 A pair of categories is used for this requirement occurrence.

EXAMPLE 2 In domain terms, when a `Via` to `Via` spacing is to be greater than 5 mm, the encoding will be that the spacing between the `reference_design_object_category` whose value is `interconnect_module_via` and the `dependent_design_object_category` whose value is `interconnect_module_via` shall be greater than 5 mm.

EXAMPLE 3 In domain terms, when the distance from a cutout to a trace must be greater than 5 mm, the encoding will be that the spacing between the `reference_design_object_category` whose value is `interconnect_module_cutout` and the `dependent_design_object_category` whose value is `interconnect_module_stratum_feature` shall be greater than 5 mm.

The data associated with a Layout_spacing_requirement_occurrence are the following:

- dependent_design_object_category;
- required_coordinated_characteristic;
- reference_design_object_category;
- of_spacing_type.

4.2.509.1 dependent_design_object_category

The dependent_design_object_category specifies one role of either an assembly_module_assembly_component, an assembly_module_component_feature, an interconnect_module_cutout, an interconnect_module_fill_area, an interconnect_module_inter_stratum_feature, an interconnect_module_stratum_feature, or an interconnect_module_via, for the Layout_spacing_requirement_occurrence.

4.2.509.2 required_coordinated_characteristic

The required_coordinated_characteristic inherited from the Ee_requirement_occurrence supertype specifies the Tolerance_characteristic for the Layout_spacing_requirement_occurrence.

4.2.509.3 reference_design_object_category

The reference_design_object_category specifies one role of either an assembly_module_assembly_component, an assembly_module_component_feature, an interconnect_module_cutout, an interconnect_module_fill_area, an interconnect_module_inter_stratum_feature, an interconnect_module_stratum_feature, or an interconnect_module_via, for the Layout_spacing_requirement_occurrence.

4.2.509.4 of_spacing_type

The of_spacing_type is either "centroid", or "nearest_boundary", or "furthest_boundary" for the Layout_spacing_requirement_occurrence.

Constraints:

UR1: The combination of the reference_design_object_category and the dependent_design_object_category shall be unique within a population of Layout_spacing_requirement_occurrences.

NOTE EXPRESS specification:

```

*)
ENTITY layout_spacing_requirement_occurrence
  SUPERTYPE OF (layer_qualified_layout_spacing_requirement_occurrence)
  SUBTYPE OF (ee_requirement_occurrence);
    reference_design_object_category : pca_product_design_object_category;
    dependent_design_object_category : pca_product_design_object_category;
    SELF\ee_requirement_occurrence.required_coordinated_characteristic :
                                     tolerance_characteristic;
    of_spacing_type                  : spacing_type;
UNIQUE
  UR1: reference_design_object_category, dependent_design_object_category;
WHERE
  WR1: 'AP210_ARM.LENGTH_DATA_ELEMENT' IN
        TYPEOF(SELF\ee_requirement_occurrence.
               required_coordinated_characteristic[1].component_characteristic[1]);
END_ENTITY; -- layout_spacing_requirement_occurrence
( *

```

4.2.510 Lead_form_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Lead_form_specification is a type of Process_specification (see 4.2.740) that describes the process to modify the shape property of a Part_terminal (see 4.2.630).

NOTE EXPRESS specification:

```

*)
ENTITY lead_form_specification
  SUBTYPE OF (process_specification);
END_ENTITY; -- lead_form_specification
( *

```

4.2.511 Length_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Length_data_element is a type of Data_element (see 4.2.211) where the measurement is of distance. A member of Length_data_element may also be a member of Qualified_data_element (see ??) in order to convey uncertainty.

NOTE EXPRESS specification:

```

*)
ENTITY length_data_element
  SUBTYPE OF (data_element);
WHERE
  WR1: SELF\data_element.measure_value &#62;= 0.0;
END_ENTITY; -- length_data_element

( *

```

4.2.512 Length_trimmed_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Length_trimmed_terminal is a type of Altered_package_terminal (see 4.2.9). A Length_trimmed_terminal describes the prepared length of a terminal accomplished prior to its assembly to the printed circuit assembly. The prepared length is equivalent to the as delivered length.

The data associated with a Length_trimmed_terminal are the following:

- maximum_trimmed_length;
- minimum_trimmed_length.

4.2.512.1 maximum_trimmed_length

The maximum_trimmed_length specifies the role of the Length_data_element (see 4.2.511) for the longest finished length of the Length_trimmed_terminal.

4.2.512.2 minimum_trimmed_length

The minimum_trimmed_length specifies the role of the Length_data_element (see 4.2.511) for the shortest finished length of the Length_trimmed_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY length_trimmed_terminal
  SUBTYPE OF (altered_package_terminal);
  maximum_trimmed_length : length_data_element;
  minimum_trimmed_length : length_data_element;
END_ENTITY; -- length_trimmed_terminal

--tt tc1 wd9 modified
( *

```

4.2.513 Library_ee_product_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Library_ee_product_definition is a type of Ee_product_definition (see 4.2.323) and a type of Library_item (see ??). A Library_ee_product_definition may be a Library_model (see ??). This provides the ability to store all design data in a library system for archival purposes.

NOTE EXPRESS specification:

```
*)
ENTITY library_ee_product_definition
-- SUPERTYPE OF (library_model)
  SUBTYPE OF (externally_defined_ee_product_definition);
DERIVE
  library : ee_name :=
    SELF\externally_defined_ee_product_definition.source;
END_ENTITY; -- library_ee_product_definition

(*
```

4.2.514 Limit_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Limit_dimension_representation is a type of Shape_dimension_representation (see 4.2.809) that explicitly describes the maximum and minimum limits of one or more Dimensional_characteristic (see 4.2.298).

The data associated with a Limit_dimension_representation are the following:

- lower_dimension_value;
- upper_dimension_value.

4.2.514.1 lower_dimension_value

The lower_dimension_value specifies the Data_element (see 4.2.211) that describes the minimum limit of the Dimensional_characteristic (see 4.2.298) for which the Limit_dimension_representation is a representation.

4.2.514.2 upper_dimension_value

The upper_dimension_value specifies the Data_element (see 4.2.211) that describes the maximum limit of the Dimensional_characteristic (see 4.2.298) for which the Limit_dimension_representation is a representation.

Constraints:

WR1: The value of the upper_dimension_value shall be greater than the value of the lower_dimension_value.

NOTE EXPRESS specification:

```

*)
ENTITY limit_dimension_representation
  SUBTYPE OF (shape_dimension_representation);
    upper_dimension_value : data_element;
    lower_dimension_value : data_element;
END_ENTITY; -- limit_dimension_representation

( *
```

4.2.515 Line_width_class

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Line_width_class is a type of Managed_design_object (see 4.2.524). A Line_width_class is a category which assigns a name to a length range for classification purposes. The Data_element used as the basis for the range shall be a Length_data_element.

EXAMPLE 1 The Line_width_class may be exchanged so that tolerances may be applied to all members of the same class, and specified only once. E.g., all lines with a design width greater than 50 mils and less than 75 mils are in the 'big' class. All members of 'big' shall have a tolerance of +5/-5 mils. Other Application Objects must be populated in order to apply the tolerances to the class.

The data associated with a Line_width_class are the following:

- class_member;
- class_name.

4.2.515.1 class_member

The class_member specifies the role of the Range_characteristic for the Line_width_class. The Range_characteristic shall provide length data.

4.2.515.2 class_name

The class_name specifies the name for the Line_width_class.

NOTE EXPRESS specification:

```

*)
ENTITY line_width_class
  SUBTYPE OF (managed_design_object);
  class_member : range_characteristic;
  class_name   : ee_name;
WHERE
  WR1: SIZEOF(QUERY( np &#60;* class_member\coordinated_characteristic.component_characteristic
    NOT('AP210_ARM.LENGTH_DATA_ELEMENT' IN TYPEOF(np\characteristic.property_value))
  )) = 0;
END_ENTITY; -- line_width_class

( *

```

4.2.516 Linear_array_placement_group_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Linear_array_placement_group_component is a type of a Array_placement_group (see 4.2.29). The placement_uncertainty relates to the originating CAD system uncertainty, not a material processing tolerance.

The data associated with a Linear_array_placement_group_component are the following:

- element;
- initial_node_positive_x_displacement;
- terminus_node_positive_x_displacement.

4.2.516.1 element

The element specifies the role of an Assembly_component_placement_link (see 4.2.38) for the Linear_array_placement_group_component. There shall be one or more Assembly_component_placement_link for a Linear_array_placement_group_component.

4.2.516.2 initial_node_positive_x_displacement

The initial_node_positive_x_displacement specifies a Length_data_element (see 4.2.511) for the Linear_array_placement_group_component. The initial_node_positive_x_displacement is the offset in the positive x direction from the local origin of the first element of the array.

4.2.516.3 terminus_node_positive_x_displacement

The terminus_node_positive_x_displacement specifies a Length_data_element (see 4.2.511) for the Linear_array_placement_group_component. The terminun_node_positive_x_displacement is the offset in the positive x direction of the last element of the array.

NOTE EXPRESS specification:

```
*)
ENTITY linear_array_placement_group_component
  SUBTYPE OF (array_placement_group);
  element : SET [1:?] OF assembly_component_placement_link;
  initial_node_positive_x_displacement : length_data_element;
  terminus_node_positive_x_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- linear_array_placement_group_component
```

(*

4.2.517 Linear_array_placement_group_component_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Linear_array_placement_group_component_link is a type of Managed_design_object (see 4.2.524).

The data associated with a Linear_array_placement_group_component_link are the following:

- precedent_element;
- referenced_by;
- subsequent_element.

4.2.517.1 precedent_element

The precedent_element specifies a Linear_array_placement_group_component (see 4.2.516) for the Linear_array_placement_group_component_link.

4.2.517.2 referenced_by

The referenced_by is an inverse relationship that specifies that the existence of the Linear_array_placement_group_component_link is dependent on the existence of the Rectangular_array_placement_group_component (see 4.2.763) that specifies the Linear_array_placement_group_component_link as its element.

4.2.517.3 subsequent_element

The subsequent_element specifies a Linear_array_placement_group_component (see 4.2.516) for the Linear_array_placement_group_component_link.

NOTE EXPRESS specification:

```
*)
ENTITY linear_array_placement_group_component_link
  SUBTYPE OF (managed_design_object);
    precedent_element : linear_array_placement_group_component;
    subsequent_element : linear_array_placement_group_component;
INVERSE
  referenced_by : rectangular_array_placement_group_component FOR
    element;
END_ENTITY; -- linear_array_placement_group_component_link

( *
```

4.2.518 Linear_extension

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Linear_extension is a type of Derived_shape_element (see 4.2.260) that is derived by linearly extending a Shape_element (see 4.2.810) that is a curve or surface. The Shape_element of which the Linear_extension is an extension is associated with the Linear_extension by the Shape_element_deriving_relationship (see 4.2.814) that are specified as the deriving_relationships.

NOTE The attribute deriving_relationships is declared in the Derived_shape_element entity of which this entity is a subtype.

Constraints:

WR1: The Linear_extension shall be derived from only one Shape_element.

IP1: The Shape_element from which the Linear_extension is derived shall be a curve or surface.

NOTE EXPRESS specification:

```
*)
ENTITY linear_extension
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- linear_extension

( *
```

4.2.519 Located_interconnect_module_thickness_requirement

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Located_interconnect_module_thickness_requirement is a type of Managed_design_object (see 4.2.524). A Located_interconnect_module_thickness_requirement is a definition that states, at a specific location, that a thickness requirement exists for an Interconnect_module (see 4.2.440).

The data associated with a Located_interconnect_module_thickness_requirement are the following:

- design;
- reference_location;
- maximum_thickness_requirement;
- minimum_thickness_requirement.

4.2.519.1 design

The design is an inverse relationship that specifies that the existence of the Located_interconnect_module_thickness_requirement is dependent on the existence of the Interconnect_module_usage_view (see 4.2.465) that specifies the Located_interconnect_module_thickness_requirement as its overall_thickness_requirement. There shall be one or more Interconnect_module_usage_view for a Located_interconnect_module_thickness_requirement.

NOTE Multiple versions of a product may have the same thickness requirement.

4.2.519.2 reference_location

The reference_location specifies the role of the Axis_placement (see 4.2.56) for the Located_interconnect_module_thickness_requirement.

4.2.519.3 maximum_thickness_requirement

The maximum_thickness_requirement specifies the Length_data_element (see 4.2.511) that is the upper bound for the Located_interconnect_module_thickness_requirement.

4.2.519.4 minimum_thickness_requirement

The minimum_thickness_requirement specifies the Length_data_element (see 4.2.511) that is the lower bound for the Located_interconnect_module_thickness_requirement.

Constraints:

UR1: The combination of the design and the reference_location shall be unique within a population of Located_interconnect_module_thickness_requirements.

NOTE EXPRESS specification:

```
*)
ENTITY located_interconnect_module_thickness_requirement
  SUBTYPE OF (managed_design_object);
    reference_location      : axis_placement;
    maximum_thickness_requirement : length_data_element;
    minimum_thickness_requirement : length_data_element;
  INVERSE
    design : SET [1:?] OF
      interconnect_module_usage_view FOR located_thickness_requirement;
  UNIQUE
    UR1 : design, reference_location;
END_ENTITY; -- located_interconnect_module_thickness_requirement

( *
```

4.2.520 Luminous_flux_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Luminous_flux_data_element is a

The data associated with a Luminous_flux_data_element are the following:

— additional_data.

4.2.520.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY luminous_flux_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- luminous_flux_data_element

( *
```

4.2.521 Luminous_intensity_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Luminous_intensity_data_element is a

The data associated with a Luminous_intensity_data_element are the following:

— additional_data.

4.2.521.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY luminous_intensity_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- luminous_intensity_data_element

( *
```

4.2.522 Magnetic_flux_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Magnetic_flux_data_element is a

The data associated with a Magnetic_flux_data_element are the following:

— additional_data.

4.2.522.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY magnetic_flux_data_element  
  SUBTYPE OF (data_element);  
END_ENTITY; -- magnetic_flux_data_element  
  
( *
```

4.2.523 Magnetic_flux_density_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Magnetic_flux_density_data_element is a

The data associated with a Magnetic_flux_density_data_element are the following:

— additional_data.

4.2.523.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY magnetic_flux_density_data_element  
  SUBTYPE OF (data_element);  
END_ENTITY; -- magnetic_flux_density_data_element  
  
( *
```

4.2.524 Managed_design_object

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Managed_design_object is one of:

- an Adjacent_stratum_surface_definition (see 4.2.5);
- an Adjacent_stratum_surface_embedded_component_surface_definition (see 4.2.4);
- an Aggregate_connectivity_requirement (see 4.2.7);
- an Alternate_select_product (see 4.2.12);
- an Analytical_model (see 4.2.14);
- an Analytical_model_port (see 4.2.17);
- an Analytical_model_port_assignment (see 4.2.18);
- an Analytical_representation (see 4.2.22);
- an Assembly_bond_definition (see 4.2.33);
- an Assembly_component (see 4.2.34);
- an Assembly_composition_relationship (see 4.2.39);
- an Assembly_joint (see 4.2.41);
- an Assembly_material_composition_relationship (see 4.2.42);
- an Assembly_requirement_allocation (see 4.2.53);
- a Bond_assembly_position (see 4.2.73);
- a Boundary_size_characteristic (see 4.2.78);
- a Bus_element_relationship (see ??);
- a Bus_structural_definition (see 4.2.83);
- a Cartesian_coordinate_system (see 4.2.86);
- a Cartesian_transformation_operator_2d (see 4.2.88);

- a Certification (see 4.2.94);
- a Characteristic (see 4.2.98);
- a Component_2d_location (see 4.2.108);
- a Component_3d_location (see 4.2.111);
- a Component_assembly_2d_position (see 4.2.112);
- a Component_assembly_3d_position (see 4.2.113);
- a Component_assembly_non_planar_position (see 4.2.114);
- a Component_feature (see 4.2.118);
- a Component_feature_make_from_relationship (see 4.2.120);
- a Component_feature_relationship (see 4.2.121);
- a Component_feature_to_layout_feature_relationship (see 4.2.122);
- a Component_feature_to_physical_usage_view_assignment (see 4.2.123);
- a Component_group_assignment (see 4.2.124);
- a Component_make_from_relationship (see 4.2.126);
- a Component_mounting_clearance_relationship (see 4.2.128);
- a Component_non_planar_location.sgm (see ??);
- a Component_overlap_relationship (see 4.2.131);
- a Component_placement_restriction_assignment (see 4.2.135);
- a Component_terminal_to_assembly_module_terminal_assignment (see ??);
- a Component_terminal_to_interconnect_module_terminal_assignment (see ??);
- a Component_terminal_to_pca_terminal_assignment (see 4.2.142);
- a Composite_curve_segment (see 4.2.156);
- a Composite_signal_property_relationship (see 4.2.164);

- a Conductive_interconnect_element_terminal_link (see 4.2.173);
- a Configuration_controlled_message (see 4.2.177);
- a Conical_tolerance_zone_boundary_and_surface_relationship (see 4.2.179);
- a Conical_tolerance_zone_boundary_diametrical_size_characteristic (see 4.2.181);
- a Connection_zone (see 4.2.185);
- a Connection_zone_model_port_assignment (see 4.2.190);
- a Connectivity_allocation (see 4.2.191);
- a Consistent_part_template_shape_projection (see ??);
- a Consistent_physical_unit_shape_projection (see ??);
- a Consistent_stratum_shape_projection (see ??);
- a Contract (see 4.2.196);
- a Coordinated_characteristic (see 4.2.197);
- a Csg_primitive (see 4.2.201);
- a Curve_style (see 4.2.205);
- a Curve_style_font (see 4.2.206);
- a Curve_style_font_and_scaling (see 4.2.207);
- a Data_element (see 4.2.211);
- a Date (see 4.2.212);
- a Datum_axis_and_tolerance_zone_orienting_relationship (see 4.2.216);
- a Datum_axis_related_orientation (see 4.2.217);
- a Datum_based_direction_vector (see 4.2.218);
- a Datum_based_length_measure (see 4.2.219);
- a Datum_based_vector_orientation (see 4.2.220);

- a Datum_feature_material_condition_property (see 4.2.227);
- a Datum_feature_usage_in_datum_system_relationship (see 4.2.230);
- a Datum_precedence_assignment (see 4.2.234);
- a Datum_system_defining_conditions (see 4.2.237);
- a Datum_target_usage_in_datum_target_set_relationship (see 4.2.242);
- a Datum_usage_in_datum_system_relationship (see 4.2.243);
- a Design_composition_path (see 4.2.262);
- a Design_functional_unit_allocation_to_assembly_component (see 4.2.263);
- a Design_functional_unit_allocation_to_assembly_joint (see 4.2.264);
- a Design_functional_unit_allocation_to_fabrication_joint (see 4.2.265);
- a Design_functional_unit_allocation_to_reference_functional_unit (see 4.2.266);
- a Design_intent_modification_in_stratum_3d_position (see ??);
- a Design_intent_modification_in_stratum_planar_position (see ??);
- a Design_intent_modification_non_planar_2d_shape (see 4.2.270);
- a Design_intent_modification_planar_shape (see 4.2.272);
- a Design_intent_modification_sequence (see 4.2.273);
- a Design_intent_modification_styled_area_representation (see 4.2.274);
- a Design_view_connection_zone_position_relationship (see 4.2.281);
- a Design_view_terminal_component_shape_relationship (see 4.2.282);
- a Device_2d_position (see 4.2.283);
- a Device_3d_position (see 4.2.284);
- a Device_terminal_map (see 4.2.285);
- a Device_terminal_map_relationship (see 4.2.286);

- a Dimension_qualifier (see 4.2.294);
- a Dimension_set (see 4.2.297);
- a Dimensional_characteristic (see 4.2.298);
- a Dimensional_characteristic_representation (see 4.2.299);
- a Direction (see 4.2.304);
- a Direction_element (see 4.2.305);
- an Ee_approval (see 4.2.314);
- an Ee_colour (see 4.2.315);
- an Ee_document (see 4.2.316);
- an Ee_document_identifier (see 4.2.317);
- an Ee_material (see 4.2.318);
- an Ee_product (see 4.2.319);
- an Ee_product_category (see 4.2.320);
- an Ee_product_category_relationship (see 4.2.321);
- an Ee_product_configuration (see 4.2.322);
- an Ee_product_definition (see 4.2.323);
- an Ee_product_model (see 4.2.325);
- an Ee_product_specific_parameter_value_assignment (see 4.2.327);
- an Ee_product_version (see 4.2.328);
- an Ee_requirement_occurrence (see 4.2.330);
- an Ee_rule_function (see 4.2.334);
- an Ee_text (see 4.2.339);
- an Engineering_make_from (see 4.2.353);

- an Equivalent_functional_terminals_assignment (see 4.2.354);
- an Equivalent_functional_unit_definition_assignment (see 4.2.355);
- an Evaluated_characteristic (see 4.2.356);
- an External_definition (see 4.2.361);
- an Externally_defined_curve_font (see 4.2.362);
- an Externally_defined_hatch_style (see 4.2.365);
- an Externally_defined_tile (see 4.2.366);
- an Externally_defined_tile_style (see 4.2.367);
- a Fabrication_joint (see 4.2.368);
- a Feature_group_relationship (see 4.2.371);
- a Fill_area (see 4.2.376);
- a Fill_area_occurrence (see 4.2.377);
- a Fill_area_style (see 4.2.378);
- a Fill_area_style_hatching (see 4.2.379);
- a Fill_area_style_tile (see 4.2.380);
- a Fill_area_style_tile_curve_with_style (see 4.2.381);
- a Fill_area_style_tile_solid_region (see 4.2.382);
- a Fill_area_style_tile_symbol_with_style (see 4.2.383);
- a Font_definition (see 4.2.387);
- a Frozen_assignment (see 4.2.394);
- a Functional_specification (see 4.2.396);
- a Functional_terminal_group (see 4.2.398);
- a Functional_terminal_group_assignment (see 4.2.399);

- a Functional_unit (see 4.2.400);
- a Functional_unit_network_node_definition (see 4.2.403);
- a Functional_unit_network_terminal_definition_bus_assignment (see 4.2.404);
- a Functional_unit_network_terminal_definition_node_assignment (see 4.2.405);
- a Functional_unit_requirement_allocation (see 4.2.406);
- a Functional_unit_terminal (see 4.2.407);
- a Functional_unit_terminal_bus_assignment (see 4.2.408);
- a Functional_unit_terminal_node_assignment (see 4.2.409);
- a Functional_unit_usage_view_terminal_definition (see 4.2.411);
- a Functional_usage_view_to_part_terminal_assignment (see 4.2.413);
- a Functional_usage_view_to_printed_part_template_terminal_assignment (see 4.2.414);
- a Geometric_representation_relationship (see 4.2.418);
- a Geometric_tolerance (see 4.2.419);
- a Geometric_tolerance_qualifier (see 4.2.420);
- an Inter_stratum_extent (see 4.2.431);
- an Inter_stratum_join_implementation (see 4.2.436);
- an Interconnect_module_product_assembly_view_fabrication_joint (see 4.2.458);
- an Interconnect_module_component_surface_feature (see 4.2.444);
- an Interconnect_module_edge_segment_surface (see 4.2.451);
- an Interconnect_module_primary_surface (see 4.2.457);
- an Interconnect_module_secondary_surface (see 4.2.459);
- an Interconnect_module_stratum_assembly_relationship (see 4.2.460);
- an Interconnect_module_surface_feature (see 4.2.462);

- an Interface_component_allocation (see 4.2.469);
- an Interface_mounted_join (see 4.2.470);
- an Internal_stratum_access (see 4.2.476);
- a Join_relationship (see 4.2.479);
- a Junction_vertex_allocation (see 4.2.482);
- a Layer (see 4.2.498);
- a Layer_connection_point (see 4.2.499);
- a Layer_connection_point_link (see 4.2.500);
- a Layout_spacing_requirement_non_conforming_design_object (see 4.2.508);
- a Line_width_class (see 4.2.515);
- a Located_interconnect_module_thickness_requirement (see 4.2.519);
- a Marking (see 4.2.526);
- a Material_composition_relationship (see 4.2.530);
- a Mating_connector_component (see 4.2.538);
- a Mating_connector_placement_relationship (see 4.2.539);
- a Mating_connector_termination (see 4.2.540);
- a Measure_qualification (see 4.2.543);
- a Model_parameter (see 4.2.549);
- a Next_higher_assembly_interface_requirement (see 4.2.555);
- an Operational_requirement_occurrence_relationship (see 4.2.567);
- an Organization (see 4.2.569);
- an Package_external_reference (see 4.2.580);
- a Package_terminal_external_reference (see 4.2.582);

- a Package_terminal_occurrence (see 4.2.583);
- a Packaged_connector_terminal_relationship (see 4.2.590);
- a Parameter_assignment (see 4.2.603);
- a Part_connected_terminals_definition (see 4.2.605);
- a Part_device (see 4.2.606);
- a Part_device_terminal (see 4.2.607);
- a Part_device_terminal_model_port_assignment (see 4.2.608);
- a Part_feature_model_port_assignment (see 4.2.611);
- a Part_template_3d_shape (see 4.2.621);
- a Part_template_non_planar_2d_shape (see 4.2.625);
- a Part_template_planar_shape (see 4.2.628);
- a Participant_date_and_time_assignment (see 4.2.637);
- a Passage_technology (see 4.2.638);
- a Person (see 4.2.646);
- a Person_and_organization (see 4.2.647);
- a Physical_connectivity_abstraction_map (see 4.2.651);
- a Physical_connectivity_definition (see 4.2.652);
- a Physical_connectivity_definition_relationship (see 4.2.653);
- a Physical_connectivity_element (see 4.2.654);
- a Physical_connectivity_make_from_relationship (see 4.2.656);
- a Physical_network (see 4.2.660);
- a Physical_node_requirement_to_implementing_component_allocation (see 4.2.663);
- a Physical_unit_3d_shape (see 4.2.666);

- a `Physical_unit_planar_shape` (see 4.2.670);
- a `Physical_unit_relationship` (see 4.2.671);
- a `Pitch_class` (see 4.2.673);
- a `Placement_group` (see 4.2.674) ;
- a `Placement_group_area_assignment` (see 4.2.675);
- a `Placement_group_volume_assignment` (see 4.2.676);
- a `Planned_characteristic` (see 4.2.679);
- a `Planned_effectivity` (see 4.2.681);
- a `Positional_boundary_and_profile_boundary_member_relationship` (see 4.2.699);
- a `Positional_boundary_definition` (see 4.2.700);
- a `Positional_boundary_member_definition` (see 4.2.702);
- a `Positional_boundary_offset` (see 4.2.703);
- a `Positive_ratio_measure` (see ??);
- a `Precision_qualification` (see ??);
- a `Printed_connector_template_terminal_relationship` (see 4.2.727);
- a `Printed_part_template_connected_terminals_definition` (see 4.2.731);
- a `Printed_part_template_material` (see 4.2.735);
- a `Printed_part_template_terminal` (see 4.2.737);
- a `Printed_part_template_terminal_relationship` (see 4.2.738);
- a `Product_association` (see 4.2.741);
- a `Projected_zone_and_base_relationship` (see 4.2.753);
- a `Projected_zone_height_characteristic` (see 4.2.754);
- a `Promissory_usage_in_product_model_relationship` (see 4.2.755);

- a Promissory_usage_relationship (see 4.2.756);
- a Reference_composition_path (see 4.2.764);
- a Reference_functional_unit_assignment_to_part (see 4.2.765);
- a Registered_font (see 4.2.768);
- a Requirement_allocation (see 4.2.769);
- a Requirement_composition (see 4.2.770);
- a Restraint_condition (see 4.2.774);
- a Rule_action (see 4.2.780);
- a Rule_justification (see 4.2.787);
- a Rule_priority (see 4.2.789);
- a Rule_set (see 4.2.792);
- a Runout_tolerance_zone_orienting_relationship (see 4.2.797);
- a Runout_zone_orientation (see 4.2.798);
- a Security_classification (see 4.2.804);
- a Separate_geometric_tolerance_requirement (see 4.2.805);
- a Shape_definition (see 4.2.808);
- a Shape_dimension_representation (see 4.2.809);
- a Shape_element (see 4.2.810);
- a Shape_element_locating_relationship (see 4.2.815);
- a Shape_element_relationship (see 4.2.816);
- a Shape_item (see 4.2.818);
- a Signal (see 4.2.823);
- a Signal_property (see 4.2.824);

- a `Signal_specification` (see 4.2.825);
- a `Simultaneous_geometric_tolerance_requirement` (see 4.2.826);
- a `Solid_curve` (see 4.2.831);
- a `Solid_fill_style` (see 4.2.832);
- a `Specification_allocation` (see 4.2.836);
- a `Stratum` (see 4.2.845);
- a `Stratum_3d_shape` (see 4.2.846);
- a `Stratum_average_surface` (see 4.2.847);
- a `Stratum_concept_relationship` (see 4.2.849);
- a `Stratum_concept_to_physical_usage_view_assignment` (see 4.2.850);
- a `Stratum_feature_3d_shape` (see 4.2.852);
- a `Stratum_feature_in_stratum_3d_position` (see 4.2.854);
- a `Stratum_feature_in_stratum_planar_position` (see 4.2.855);
- a `Stratum_feature_non_planar_2d_shape` (see 4.2.856);
- a `Stratum_feature_shape_stratum_average_surface_shape_relationship` (see 4.2.858);
- a `Stratum_feature_shape_stratum_surface_shape_relationship` (see 4.2.859);
- a `Stratum_feature_styled_area_representation` (see 4.2.860);
- a `Stratum_interconnect_module_3d_position` (see 4.2.863);
- a `Stratum_interconnect_module_planar_position` (see 4.2.864);
- a `Stratum_make_from_relationship` (see 4.2.865);
- a `Stratum_planar_shape` (see 4.2.867);
- a `Stratum_surface` (see 4.2.868);
- a `Stratum_surface_in_stratum_3d_position` (see 4.2.869);

- a Stratum_technology (see 4.2.871);
- a Structured_text (see 4.2.874);
- a Styled_curve_occurrence (see 4.2.875);
- a Sub_assembly_relationship (see 4.2.877);
- a Substitute_product (see 4.2.878);
- a Supplied_product_version (see 4.2.879);
- a Technology (see 4.2.886);
- a Template_stratum_assignment (see 4.2.889);
- a Terminal_pair_port_equivalence_relationship (see 4.2.890);
- a Termination_constraint (see 4.2.892);
- a Thermal_isolation_requirement_assignment (see 4.2.905);
- a Tolerance_zone_and_viewing_plane_orienting_relationship (see 4.2.914);
- a Tolerance_zone_boundary_definition (see 4.2.916);
- a Tolerance_zone_boundary_member_offset (see 4.2.918);
- a Tolerance_zone_definition (see 4.2.920);
- a Tolerance_zone_form (see 4.2.924);
- a Tolerance_zone_per_unit_size_specification (see 4.2.927);
- a Topological_junction (see 4.2.928);
- a Transmission_line_functional_unit_terminal_allocation (see 4.2.931);
- a True_profile (see 4.2.934);
- a Type_qualifier (see 4.2.935);
- a Uncertainty_qualifier (see 4.2.936);
- an Usage_concept_usage_relationship (see 4.2.942);

- a Vector (see 4.2.946);
- a Viewing_plane_based_orientation (see 4.2.957);
- a Viewing_plane_definition (see 4.2.958);
- a Volume_shape_projection (see ??);
- a Work_order (see 4.2.964);
- a Work_request (see 4.2.965).

Additionally, a Managed_design_object may be a Frozen_design_object (see 4.2.395). Additionally, a Managed_design_object may be a Identified_design_object (see ??). Additionally, a Managed_design_object may be a Owned_design_object (see 4.2.570). A Managed_design_object is the explicit subject of a change specification.

NOTE EXPRESS specification:

*)

ENTITY managed_design_object

ABSTRACT SUPERTYPE OF ((ONEOF (adjacent_stratum_surface_definition,
adjacent_stratum_surface_embedded_component_surface_definition,
alternate_select_product,
analytical_model, analytical_model_make_from_relationship,
analytical_model_port, analytical_model_port_assignment,
analytical_representation,
assembly_bond_definition, assembly_component,
assembly_component_placement_link,
assembly_composition_relationship, assembly_joint,
assembly_material_composition_relationship,
assembly_requirement_allocation,
bond_assembly_position,
boundary_size_characteristic, bus_element_link,
bus_structural_definition, cartesian_coordinate_system,
cartesian_transformation_operator_2d, certification,
characteristic, component_2d_location, component_3d_location,
component_assembly_2d_position, component_assembly_3d_position,
component_assembly_non_planar_position,
component_external_reference, component_feature,
component_feature_external_reference,
component_feature_make_from_relationship,
component_feature_relationship,
component_feature_to_layout_feature_relationship,
component_feature_to_physical_usage_view_assignment,
component_group_assignment, component_make_from_relationship,
component_mounting_clearance_relationship,
component_non_planar_location,
component_overlap_relationship,
component_placement_restriction_assignment,
composite_curve_segment, composite_signal_property_relationship,
conductive_interconnect_element_terminal_link,
configuration_controlled_message,
conical_tolerance_zone_boundary_and_surface_relationship,
conical_tolerance_zone_boundary_diametrical_size_characteristic,
connection_zone, connectivity_allocation,
connectivity_allocation_to_physical_network,
connection_zone_interface_plane_relationship,
connection_zone_map_identification,
connection_zone_model_port_assignment,
coordinated_characteristic, csg_primitive, curve_style,
curve_style_font, curve_style_font_and_scaling, data_element, date,
datum_axis_and_tolerance_zone_orienting_relationship,
datum_axis_related_orientation, datum_based_direction_vector,
datum_based_length_measure, datum_based_vector_orientation,
datum_difference_analytical_model_port_assignment,
datum_difference_functional_unit_usage_view_terminal_assignment,
datum_feature_material_condition_property,

datum_feature_usage_in_datum_system_relationship,
 datum_precedence_assignment, datum_system_defining_conditions,
 datum_target_usage_in_datum_target_set_relationship,
 datum_usage_in_datum_system_relationship,
 design_composition_path,
 design_functional_unit_allocation_to_assembly_component,
 design_functional_unit_allocation_to_assembly_joint,
 design_functional_unit_allocation_to_fabrication_joint,
 design_functional_unit_allocation_to_reference_functional_unit,
 design_intent_modification,
 design_intent_modification_sequence,
 design_intent_modification_3d_position,
 design_intent_modification_planar_position,
 design_intent_modification_styled_area_representation,
 design_view_connection_zone_position_relationship,
 design_view_terminal_component_shape_relationship,
 device_2d_position, device_3d_position, device_terminal_map,
 device_terminal_map_relationship, dimensional_characteristic,
 dimensional_characteristic_representation, dimension_qualifier,
 dimension_set, direction, direction_element,
 ee_approval, ee_colour, ee_document,
 ee_document_identifier, ee_material, ee_product, ee_product_category,
 ee_product_category_relationship, ee_product_configuration,
 ee_product_definition, ee_product_model,
 ee_product_specific_parameter_value_assignment, ee_product_version,
 ee_requirement_occurrence,
 ee_rule_complex_clause,
 ee_rule_conclusion_definition,
 ee_rule_function,
 ee_rule_premise_definition,
 ee_rule_simple_clause,
 ee_text, engineering_make_from,
 equivalent_functional_terminals_assignment,
 equivalent_functional_unit_definition_assignment,
 evaluated_characteristic, externally_defined_curve_font,
 externally_defined_hatch_style, externally_defined_tile,
 externally_defined_tile_style, external_definition, fabrication_joint,
 feature_group_relationship,
 fill_area,
 fill_area_occurrence, fill_area_style, fill_area_style_hatching,
 fill_area_style_tile, fill_area_style_tile_curve_with_style,
 fill_area_style_tile_solid_region,
 fill_area_style_tile_symbol_with_style, font_definition,
 frozen_assignment, functional_specification,
 functional_terminal_group, functional_terminal_group_assignment,
 functional_unit, functional_unit_network_node_definition,
 functional_unit_network_terminal_definition_bus_assignment,
 functional_unit_network_terminal_definition_node_assignment,
 functional_unit_requirement_allocation, functional_unit_terminal,
 functional_unit_terminal_bus_assignment,
 functional_unit_terminal_node_assignment,
 functional_unit_usage_view_terminal_definition,

functional_usage_view_to_part_terminal_assignment,
 functional_usage_view_to_printed_part_template_terminal_assignment,
 geometric_tolerance,geometric_tolerance_qualifier,
 interconnect_module_product_assembly_view_fabrication_joint,
 interconnect_module_stratum_assembly_relationship,
 interface_component_allocation,interface_mounted_join,
 internal_stratum_access,inter_stratum_extent,
 inter_stratum_join_implementation,join_relationship,
 junction_vertex_allocation,layer,
 layer_connection_point,
 layer_connection_point_link,
 layout_spacing_requirement_non_conforming_design_object,
 line_width_class,linear_array_placement_group_component_link,
 located_interconnect_module_thickness_requirement,marking,
 material_composition_relationship,
 mating_connector_component,mating_connector_placement_relationship,
 mating_connector_termination,model_parameter,
 next_higher_assembly_interface_requirement,
 operational_requirement_occurrence_relationship,organization,
 package_external_reference,
 package_terminal_external_reference,package_terminal_occurrence,
 packaged_connector_terminal_relationship,parameter_assignment,
 participant_date_and_time_assignment,
 part_connected_terminals_definition,part_device,
 part_device_terminal,part_device_terminal_model_port_assignment,
 part_feature_model_port_assignment,
 part_terminal_external_reference,
 part_template_or_physical_unit_2d_shape_representation_relationship,
 part_template_or_physical_unit_3d_shape_representation_relationship,
 passage_technology,person,
 person_and_organization,physical_connectivity_abstraction_map,
 physical_connectivity_definition,
 physical_connectivity_definition_relationship,
 physical_connectivity_element,
 physical_connectivity_make_from_relationship,physical_network,
 physical_node_requirement_to_implementing_component_allocation,
 physical_unit_relationship,pitch_class,placement_group,
 placement_group_area_assignment,planned_characteristic,
 planned_effectivity,
 positional_boundary_and_profile_boundary_member_relationship,
 positional_boundary_definition,
 positional_boundary_member_definition,positional_boundary_offset,
 printed_connector_template_terminal_relationship,
 printed_part_template_connected_terminals_definition,
 printed_part_template_material,printed_part_template_material_link,
 printed_part_template_terminal,
 printed_part_template_terminal_relationship,product_association,
 projected_zone_and_base_relationship,
 projected_zone_height_characteristic,promissory_usage_relationship,
 promissory_usage_in_product_model_relationship,reference_composition_path,
 reference_functional_unit_assignment_to_part,registered_font,
 requirement_allocation,

```

requirement_composition, restraint_condition, rule_action,
rule_justification, rule_priority, rule_set,
runout_tolerance_zone_orienting_relationship,
runout_zone_orientation, security_classification,
scalar_terminal_definition_link,
separate_geometric_tolerance_requirement,
shape_definition,
shape_dimension_representation, shape_element,
shape_element_locating_relationship, shape_element_relationship,
shape_item,
signal, signal_property, signal_specification,
simultaneous_geometric_tolerance_requirement, solid_curve,
solid_fill_style, specification_allocation,
stratum_concept_relationship,
stratum_concept_to_physical_usage_view_assignment,
stratum_feature_in_stratum_3d_position,
stratum_feature_in_stratum_planar_position,
stratum_feature_shape_stratum_average_surface_shape_relationship,
stratum_feature_shape_stratum_surface_shape_relationship,
stratum_feature_styled_area_representation,
stratum_interconnect_module_3d_position,
stratum_interconnect_module_planar_position,
stratum_make_from_relationship,
stratum_average_surface,
stratum_surface, stratum_surface_in_stratum_3d_position,
stratum_technology,
stratum_technology_link, structured_text,
styled_curve_occurrence, substitute_product,
sub_assembly_relationship, supplied_product_version, technology,
template_stratum_assignment,
terminal_pair_port_equivalence_relationship, termination_constraint,
thermal_isolation_requirement_assignment,
tolerance_zone_and_viewing_plane_orienting_relationship,
tolerance_zone_boundary_definition,
tolerance_zone_boundary_member_offset, tolerance_zone_definition,
tolerance_zone_form, tolerance_zone_per_unit_size_specification,
topological_junction,
transmission_line_functional_unit_terminal_allocation, true_profile,
usage_concept_usage_relationship,
usage_view_connection_zone_terminal_shape_relationship,
vector,
viewing_plane_based_orientation, viewing_plane_definition,
volume_shape_intersection, volume_shape_plane_projection,
volume_shape_point_projection,
work_order, work_request) ANDOR
frozen_design_object) ANDOR
owned_design_object);
END_ENTITY; -- managed_design_object
( *

```

4.2.525 Manifold_subsurface_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Manifold_subsurface_shape` is a type of `shape_definition` (see 4.2.808).

NOTE EXPRESS specification:

```
*)
ENTITY manifold_subsurface_shape
  SUBTYPE OF (shape_definition);
END_ENTITY; -- manifold_subsurface_shape
```

(*

4.2.526 Marking

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Marking is a type of `Managed_design_object` (see 4.2.524). A Marking is the specification of what, where, and how information that is placed on a `Physical_unit` (see 4.2.664) for identification purposes is applied. Marking in this part of ISO 10303 is considered to not be part of the definition of a `Stratum` (see 4.2.845). Marking is text that is applied to a material object. The `message_representation_text` shall be an item in the `message_representation`.

EXAMPLE 1 `Laminate_text_component` is text that is realized through the manufacturing process. Reference designators that are created on the board during etching are members of `Laminate_text_component`.

The data associated with a Marking are the following:

- `marked_device_shape`;
- `marked_message`;
- `marking_material`;
- `marking_specification`;
- `message_representation`;
- `message_representation_text`;
- `placement`.

4.2.526.1 marked_device_shape

The `marked_device_shape` specifies either a `Physical_unit_3d_shape` (see 4.2.666), or a `Physical_unit_planar_shape` (see 4.2.670) that is the shape definition for the physical device that receives the Marking.

4.2.526.2 marked_message

The marked_message specifies either a Configuration_controlled_message (see 4.2.177) or Structured_text (see 4.2.874) for Marking.

4.2.526.3 marking_material

The marking_material specifies the Ee_material (see 4.2.318) used in the Marking.

4.2.526.4 marking_specification

The marking_specification specifies the Ee_specification (see 4.2.338) of the process and format for properly applying the Marking.

4.2.526.5 message_representation

The message_representation specifies role of the Feature_shape_definition (see 4.2.372) for the Marking.

4.2.526.6 message_representation_text

The message_representation_text specifies role of the Styled_text (see 4.2.876) for the Marking.

4.2.526.7 placement

The placement specifies the Axis_placement (see 4.2.56) of the Marking. The placement specifies the placement of the message_representation in the marked_device_shape.

NOTE EXPRESS specification:

```

*)
ENTITY marking
  SUBTYPE OF (managed_design_object);
    marked_message      : marked_message_select;
    marked_device_shape : view_shape_select;
    marking_material    : ee_material;
    placement           : axis_placement;
    message_representation : feature_shape_definition;
    message_representation_text : styled_text;
    marking_specification : ee_specification;
WHERE
  WR1: SIZEOF(QUERY(st &#60;*
    message_representation\shape_definition.element |
    st = message_representation_text)) = 1;
  WR2: SIZEOF(QUERY(ap &#60;*
    marked_device_shape\shape_definition.element |
    ap = placement)) = 1;
  WR3: SIZEOF(TYPEOF(marked_device_shape) *
    ['AP210_ARM.PHYSICAL_UNIT_3D_SHAPE',
    'AP210_ARM.PHYSICAL_UNIT_PLANAR_SHAPE'])= 1;
END_ENTITY; -- marking

( *
```

4.2.527 Mass_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mass_data_element is a

The data associated with a Mass_data_element are the following:

— additional_data.

4.2.527.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY mass_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- mass_data_element

( *
```

4.2.528 Material_addition_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_addition_feature is a type of Design_intent_modification (see 4.2.267) that describes the material that is to be added in the realization of the feature specified by the design_intent attribute inherited from Design_intent_modification. This requirement shall be interpreted as being mathematically equivalent to the application of the regularized Boolean geometric OR operation where the design_intent is OR'd with the shape of the Material_addition_feature.

NOTE EXPRESS specification:

```
*)
ENTITY material_addition_feature
  SUBTYPE OF (design_intent_modification);
  material_addition_shape : material_addition_feature_template;
END_ENTITY; -- material_addition_feature

( *
```

4.2.529 Material_addition_feature_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_addition_feature_template is a type of Part_template (see 4.2.620) that describes the shape of material to be added to a design feature.

The data associated with a Material_addition_feature_template are the following:

— of_stratum_technology.

4.2.529.1 of_stratum_technology

The of_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Material_addition_feature_template.

NOTE EXPRESS specification:

```

*)
ENTITY material_addition_feature_template
  SUBTYPE OF (part_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- material_addition_feature_template

( *

```

4.2.530 Material_composition_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_composition_relationship is a type of Managed_design_object (see 4.2.524). A Material_composition_relationship is the association between one Ee_material (see 4.2.318) and another when describing a composite material.

The data associated with a Material_composition_relationship are the following:

- composition_basis;
- composition_class;
- constituent_amount;
- determination_method;
- related_product;
- relating_product.

4.2.530.1 composition_basis

The composition_basis specifies the string that represents the basis on which a material product is decomposed into its constituents.

4.2.530.2 composition_class

The composition_class specifies the string that represents the category of composition for the Material_composition_relationship.

4.2.530.3 constituent_amount

The constituent_amount specifies the Data_element (see 4.2.211) that represents the amount of material.

4.2.530.4 determination_method

The `determination_method` specifies the string that represents the procedure by which `constituent_amount` is determined.

4.2.530.5 related_product

The `related_product` is `Ee_material` (see 4.2.318) that is the constituent for the `Material_composition_relationship`. There shall be one or more `Ee_material` for the `Material_composition_relationship`.

4.2.530.6 relating_product

The `relating_product` is `Ee_material` (see 4.2.318) that is the composition for which this `Material_composition_relationship` is specifying a constituent.

NOTE This relationship is intended for compounds and simple mixtures only, not for assemblies.

NOTE EXPRESS specification:

```

*)
ENTITY material_composition_relationship
  SUBTYPE OF (managed_design_object);
    composition_basis      : STRING;
    determination_method  : ee_text;
    composition_class      : STRING;
    constituent_amount     : data_element;
    related_product        : ee_material;
    relating_product       : ee_material;
END_ENTITY; -- material_composition_relationship

```

(*

4.2.531 Material_definition

A `Material_definition` is a type of `Ee_product_definition` (see 4.2.323). A `Material_definition` is intended solely to capture a minimal definition and identification of raw material for bill of material or product data purposes.

The data associated with a `Material_definition` are the following:

— `material`.

4.2.531.1 material

The `material` specifies role of the `Ee_material` (see 4.2.318) for the `Material_definition`.

NOTE EXPRESS specification:

```
*)
ENTITY material_definition
  SUBTYPE OF (ee_product_definition);
  material: ee_material;
END_ENTITY; -- material_definition
(*
```

4.2.532 Material_electrical_conductivity_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_electrical_conductivity_requirement_occurrence is a type of Electrical_requirement_occurrence (see 4.2.346) that specifies the requirement for conductive material.

The data associated with a Material_electrical_conductivity_requirement_occurrence are the following:

— electrical_conductivity_characteristic.

4.2.532.1 electrical_conductivity_characteristic

The electrical_conductivity_characteristic specifies either a conductive, a non_conductive, a resistive, or a semi_conductive classification of material conductivity for the Material_electrical_conductivity_classification.

NOTE These are generic category names. If more specific data is required, the characteristics should be associated to the Material_electrical_conductivity_requirement_occurrence using the required_coordinated_characteristic or the required_characteristic inherited from Ee_requirement_occurrence.

NOTE EXPRESS specification:

```
*)
ENTITY material_electrical_conductivity_requirement_occurrence
  SUBTYPE OF (electrical_requirement_occurrence);
  electrical_conductivity_characteristic : material_conductivity_classification;
END_ENTITY; -- material_electrical_conductivity_requirement_occurrence
(*
```

4.2.533 Material_removal_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Material_removal_feature` is a type of `Design_intent_modification` (see 4.2.267) that indicates that material is to be removed in the realization of the feature that is in the role of `design_intent` inherited from `Design_intent_modification`. A `Material_removal_feature` may be a `Filled_area_material_removal_structure` (see 4.2.385). The requirement shall be interpreted mathematically as a requirement for application of the regularized Boolean subtraction operation.

The data associated with a `Material_removal_feature` are the following:

— `material_removal_shape`.

4.2.533.1 `material_removal_shape`

The `material_removal_shape` specifies the role of the `Material_removal_feature_template` (see 4.2.534) for the `Material_removal_feature`.

NOTE EXPRESS specification:

```
*)
ENTITY material_removal_feature
  SUPERTYPE OF (filled_area_material_removal_structure)
  SUBTYPE OF (design_intent_modification);
  material_removal_shape : material_removal_feature_template;
END_ENTITY; -- material_removal_feature

(*
```

4.2.534 `Material_removal_feature_template`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

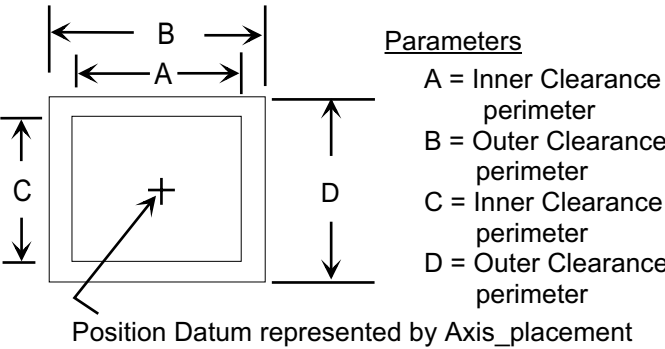
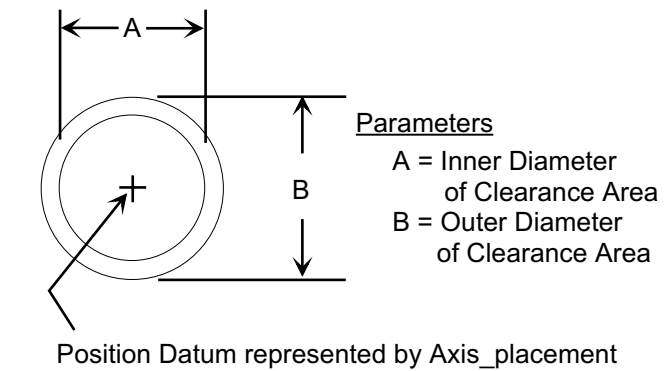
A `Material_removal_feature_template` is a type of `Part_template` (see 4.2.620) that describes the shape of material to be removed from a feature. A `Material_removal_feature_template` may be an `Electrical_isolation_removal_template` (see 4.2.344) or a `Thermal_isolation_removal_template` (see 4.2.904).

EXAMPLE 1 Figure 108 illustrates a template and associated parameters designed for electrical isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values.

EXAMPLE 2 Figure 109 illustrates a template and associated parameters designed for thermal isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values. In this example, an instance would be a complex instance including `Composite_unit_shape_element` (see 4.2.167) and `Material_removal_feature_template`.

The data associated with a `Material_removal_feature_template` are the following:

— `of_stratum_technology`.

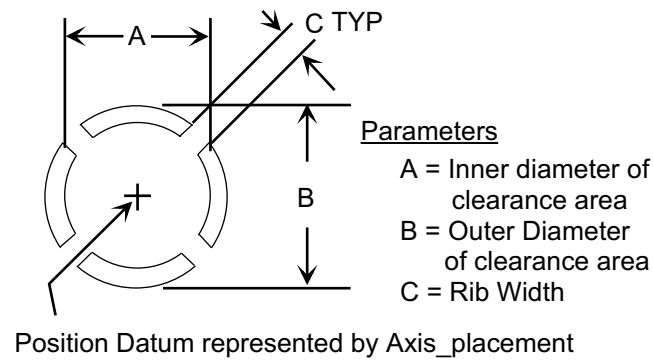
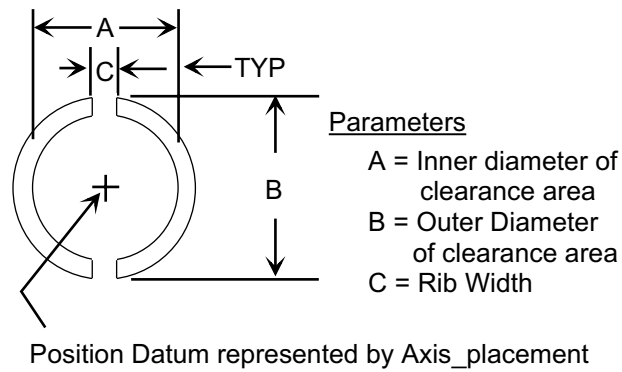


Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Electrical Isolation

Figure 61 – Material_removal_feature_template supports electrical isolation



Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Thermal isolation

Figure 62 – Material_removal_feature_template supports thermal isolation and electrical conduction

4.2.534.1 of_stratum_technology

The of_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Material_removal_feature_template.

NOTE EXPRESS specification:

```

*)
ENTITY material_removal_feature_template
  SUPERTYPE OF (ONEOF(electrical_isolation_removal_template,
    thermal_isolation_removal_template))
  SUBTYPE OF (part_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- material_removal_feature_template

( *
```

4.2.535 Material_removal_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_removal_laminate_component is a type of Laminate_component (see 4.2.483) that indicates that material is to be removed in the realization of the feature that is in the role of design_intent. A Material_removal_laminate_component may be a Filled_area_material_removal_structure (see 4.2.385). The requirement shall be interpreted mathematically as a requirement for application of the regularized Boolean subtraction operation. A Material_removal_laminate_component may participate in a relationship with a Design_intent_modification_sequence (see 4.2.273) in the case of complex layout arrangements to resolve the order of the Boolean operations.

The data associated with a Material_removal_laminate_component are the following:

- design_intent;
- usage_definition.

4.2.535.1 design_intent

The design_intent specifies the role of the Stratum_feature (see 4.2.851) for the Material_removal_laminate_component.

4.2.535.2 usage_definition

The usage_definition specifies the role of the Material_removal_feature_template (see 4.2.534) for the Material_removal_laminate_component.

NOTE EXPRESS specification:

```

*)
ENTITY material_removal_laminate_component
  SUPERTYPE OF (filled_area_material_removal_laminate_component)
  SUBTYPE OF (laminate_component);
  SELF\assembly_component.usage_definition :
    material_removal_feature_template;
  design_intent : stratum_feature;
END_ENTITY; -- material_removal_laminate_component

( *

```

4.2.536 Material_shape_modification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_shape_modification is a type of Design_intent_modification (see 4.2.267) that indicates that the shape of material of the feature in the role of design_intent inherited from Design_intent_modification is to be modified to suit the design intent. The mathematical interpretation shall be dependent on the operator instance data in the geometrical representation used in the model. A Material_shape_modification shall be represented only by a geometric representation capable of supporting regularized Boolean operations.

NOTE EXPRESS specification:

```

*)
ENTITY material_shape_modification
  SUBTYPE OF (design_intent_modification);
END_ENTITY; -- material_shape_modification

( *

```

4.2.537 Material_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Material_specification is a type of Ee_specification (see 4.2.338) that specifies properties applicable to raw material, mixtures, or semi-fabricated material that are used in the fabrication of a product. A Material_specification may be a Surface_finish_specification (see 4.2.880).

EXAMPLE 1 An example of a semi-fabricated material is the partially cured adhesive sheets which often are composed of a glass fiber mat and "B-stage" epoxy. Semi-fabricated material may be purchased or sold by an enterprise, in the same way that raw materials or mixtures may.

NOTE EXPRESS specification:

```

*)
ENTITY material_specification
  SUPERTYPE OF (surface_finish_specification)
  SUBTYPE OF (ee_specification);
END_ENTITY; -- material_specification

( *

```

4.2.538 Mating_connector_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mating_connector_component is a type of Managed_design_object (see 4.2.524). A Mating_connector_component is the connector on the next higher assembly with which a connector on the assembly must mate. A Mating_connector_component may be specified in the role of mating_requirement by an Interface_component_allocation (see 4.2.469).

The data associated with a Mating_connector_component are the following:

- definition;
- placement_context;
- reference_designation.

4.2.538.1 definition

The definition specifies the role of either an Ee_specification (see 4.2.338), a Packaged_connector (see 4.2.587), or a Printed_connector_template (see 4.2.726) for the Mating_connector_component.

4.2.538.2 placement_context

The placement_context is an inverse relationship that specifies that the existence of the Mating_connector_component is dependent on the existence of the Mating_connector_placement_relationship (see 4.2.539) that specifies the Mating_connector_component as its placed_connector.

4.2.538.3 reference_designation

The reference_designation specifies the name that serves as the identifier of the location of the Mating_connector_component in the next level of assembly. The reference_designation need not be specified for a particular Mating_connector_component.

NOTE EXPRESS specification:

```

*)
ENTITY mating_connector_component
  SUBTYPE OF (managed_design_object);
  definition          : mating_connector_part;
  reference_designation : OPTIONAL ee_name;
  INVERSE
    placement_context : mating_connector_placement_relationship FOR
                        placed_connector;
END_ENTITY; -- mating_connector_component

( *

```

4.2.539 Mating_connector_placement_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mating_connector_placement_relationship is a type of Managed_design_object (see 4.2.524). A Mating_connector_placement_relationship is the association between the connector that the interface connector on the assembly must mate with in the next higher assembly, and the resultant placement of that mating connector in the assembly being designed for test and evaluation. The evaluation may include fit checks and evaluation of other material properties. The application of Mating_connector_placement_relationship is limited to three-dimensional geometric systems.

The data associated with a Mating_connector_placement_relationship are the following:

- connector_placement;
- interface_context;
- placed_connector.

4.2.539.1 connector_placement

The connector_placement specifies the role of the Axis_placement (see 4.2.56) for the Mating_connector_placement_relationship. The Axis_placement shall be an Axis2_placement_3d.

4.2.539.2 interface_context

The interface_context specifies the role of the Next_higher_assembly_interface_requirement (see 4.2.555) for the Mating_connector_placement_relationship. The coordinated system of the shape representing the Next_higher_assembly_interface_requirement shall be congruent with the coordinate system of the assembly being designed.

4.2.539.3 placed_connector

The placed_connector specifies the role of the Mating_connector_component (see 4.2.538) for the Mating_connector_placement_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY mating_connector_placement_relationship
  SUBTYPE OF (managed_design_object);
    placed_connector      : mating_connector_component;
    interface_context     : next_higher_assembly_interface_requirement;
    connector_placement   : axis_placement;
END_ENTITY; -- mating_connector_placement_relationship

( *
```

4.2.540 Mating_connector_termination

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mating_connector_termination is a type of Managed_design_object (see 4.2.524). A Mating_connector_termination is a single terminal of a Mating_connector_component.

The data associated with a Mating_connector_termination are the following:

- accessed_connector;
- definition;
- name;
- signal_definition.

4.2.540.1 accessed_connector

The accessed_connector specifies the role of the Mating_connector_component (see 4.2.538) for the Mating_connector_termination.

4.2.540.2 definition

The definition specifies either a Packaged_part_terminal (see 4.2.594) or a Printed_part_template_terminal (see 4.2.737) for the Mating_connector_termination.

4.2.540.3 name

The name specifies the identifier for the Mating_connector_termination.

4.2.540.4 signal_definition

The signal_definition specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Mating_connector_termination. The signal_definition need not be specified for a particular Mating_connector_termination.

Constraints:

UR1: The combination of the accessed_connector and the definition shall be unique within a population of Mating_connector_termination.

NOTE EXPRESS specification:

```

*)
ENTITY mating_connector_termination
  SUBTYPE OF (managed_design_object);
    definition      : packaged_or_printed_terminal;
    accessed_connector : mating_connector_component;
    name            : ee_name;
    signal_definition : OPTIONAL ee_requirement_occurrence;
  UNIQUE
    UR1 : definition, accessed_connector;
END_ENTITY; -- mating_connector_termination

( *
```

4.2.541 Maximum_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Maximum_dimension_representation is a type of Singular_dimension_representation (see 4.2.829) that explicitly describes the maximum limit of one or more Dimensional_characteristic (see 4.2.298) and in which the minimum limit is not specified as it is determined by other elements of the design.

NOTE Maximum dimensions are discussed in 4.4 of ISO 406.

NOTE EXPRESS specification:

```
*)
ENTITY maximum_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- maximum_dimension_representation

( *
```

4.2.542 Maximum_tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Maximum_tolerance_characteristic is a type of Tolerance_characteristic (see 4.2.911). The Maximum_tolerance_characteristic indicates the maximum allowed value for the parameter specified.

The data associated with a Maximum_tolerance_characteristic are the following:

— component_characteristic.

4.2.542.1 component_characteristic

The component_characteristic specifies the maximum value of the Numeric_parameter (see 4.2.565) for the Maximum_tolerance_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY maximum_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
  SELF\coordinated_characteristic.component_characteristic :
SET [1:1] OF numeric_parameter;
END_ENTITY; -- maximum_tolerance_characteristic

( *
```

4.2.543 Measure_qualification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Measure_qualification is a type of Managed_design_object (see 4.2.524) in which there is an associated uncertainty. The uncertainty may be specified as precision, qualitative uncertainty, or expanded uncertainty, and the type of the related datum shall be included. The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY measure_qualification
  SUBTYPE OF (managed_design_object);
    name : text;
    description : text;
    qualified_measure : data_element;
    qualifiers : SET [1:?] OF value_qualifier;
END_ENTITY; -- measure_qualification
(*
```

4.2.544 Mechanical_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mechanical_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that specifies a mechanical domain related requirement.

NOTE EXPRESS specification:

```
*)
ENTITY mechanical_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- mechanical_requirement_occurrence
(*
```

4.2.545 Minimally_defined_bare_die_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Minimally_defined_bare_die_terminal is a type of Part_terminal (see 4.2.630). A Minimally_defined_bare_die_terminal may be either a Guided_wave_bare_die_terminal (see 4.2.426) or a Bare_die_terminal (see 4.2.66). The associated_definition specifies the Bare_die (see 4.2.59) to which the Bare_die_terminal provides electrical access. A Minimally_defined_bare_die_terminal does not include location or shape information

NOTE EXPRESS specification:

```

*)
ENTITY minimally_defined_bare_die_terminal
  SUPERTYPE OF (guided_wave_bare_die_terminal ANDOR bare_die_terminal)
  SUBTYPE OF (part_terminal);
  SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- minimally_defined_bare_die_terminal

( *

```

4.2.546 Minimally_defined_component_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Minimally_defined_component_terminal` is a type of `Component_terminal` (see 4.2.137) that is included to permit references to terminals without including geometric representations of the terminals.

The data associated with a `Minimally_defined_component_terminal` are the following:

— definition.

4.2.546.1 definition

The definition specifies the role of the `Part_terminal_external_reference` (see 4.2.631) for the `Minimally_defined_component_terminal`.

NOTE EXPRESS specification:

```

*)
ENTITY minimally_defined_component_terminal
  SUBTYPE OF (physical_component_terminal);
  SELF\component_feature.definition : part_terminal_external_reference;
END_ENTITY; -- minimally_defined_component_terminal

( *

```

4.2.547 Minimum_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Minimum_dimension_representation` is a type of `Singular_dimension_representation` (see 4.2.829) that explicitly describes the minimum limit of one or more `Dimensional_characteristic` (see 4.2.298) and in which the maximum limit is not specified as it is determined by other elements of the design.

NOTE Minimum dimensions are discussed in 4.4 of ISO 406.

NOTE EXPRESS specification:

```
*)
ENTITY minimum_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- minimum_dimension_representation

( *
```

4.2.548 Minimum_tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Minimum_tolerance_characteristic is a type of Tolerance_characteristic (see 4.2.911). The Minimum_tolerance_characteristic indicates the minimum allowed value for the parameter specified.

The data associated with a Minimum_tolerance_characteristic are the following:

— component_characteristic.

4.2.548.1 component_characteristic

The component_characteristic specifies the minimum value of the Numeric_parameter (see 4.2.565) for the Minimum_tolerance_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY minimum_tolerance_characteristic

  SUBTYPE OF (tolerance_characteristic);
  SELF\coordinated_characteristic.component_characteristic :
  SET [1:1] OF numeric_parameter;
END_ENTITY; -- minimum_tolerance_characteristic

( *
```

4.2.549 Model_parameter

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Model_parameter is a type of Managed_design_object (see 4.2.524). A Model_parameter is a formally declared variable of the Analytical_model (see 4.2.14) provided for an external application to populate at execution time in a computing environment. The base type of the variable is defined by the base type of the default_value or valid_range data.

EXAMPLE 1 In Spice, temperature is a Model_parameter that may be set at the execution time.

The data associated with a Model_parameter are the following:

- default_value;
- reference_document;
- type_name;
- unit_of_measure;
- valid_range.

4.2.549.1 default_value

The default_value specifies either a Characteristic (see 4.2.98) or a Coordinated_characteristic (see 4.2.197) for the parameter. The default_value need not be specified for a particular Model_parameter. When a collection of data types needs to be defined, so that an instance of the collection may be referenced, the Coordinated_characteristic is used. The External_definition (see 4.2.361) specifies the reference_document to define the semantics of the collection.

4.2.549.2 reference_document

The reference_document specifies either an Ee_document (see 4.2.316) or an identifier for the standard document that contains the reference for the Model_parameter. The reference_document shall define the mapping between the parameter_type string values and the semantics to be applied by a simulator, except for the case of physical_property_type which is predefined in this part of ISO 10303.

4.2.549.3 type_name

The type_name specifies the string used as the human-interpretable type name for the Model_parameter.

4.2.549.4 unit_of_measure

The unit_of_measure specifies the string used as the descriptive label for the unit of measure associated with the Model_parameter. The unit_of_measure need not be specified for a particular Model_parameter. The representation of units described in ISO 10303-41 shall be used.

4.2.549.5 valid_range

The valid_range specifies the Coordinated_characteristic (see 4.2.197) of the Model_parameter. The valid_range need not be specified for a particular Model_parameter. There may be more than one Coordinated_characteristic for a Model_parameter. The valid_ranges need not be contiguous.

NOTE The base units of the valid_range and default_value are the same type.

Constraints:

UR1: The combination of the reference_document and the type_name shall be unique within a population of Model_parameter.

NOTE EXPRESS specification:

```

*)
ENTITY model_parameter
  SUPERTYPE OF (datum_difference_based_model_parameter)
  SUBTYPE OF (managed_design_object);
  type_name          : STRING;
  unit_of_measure    : OPTIONAL STRING;
  reference_document : ee_document;
  valid_range        : OPTIONAL SET [1:?] OF coordinated_characteristic;
  default_value      : OPTIONAL scalar_or_coordinated_characteristics;
UNIQUE
  UR1: type_name, reference_document;
END_ENTITY; -- model_parameter

( *
```

4.2.550 Mounting_restriction_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Mounting_restriction_area is a type of Non_feature_shape_element (see 4.2.560). A Mounting_restriction_area is the region on an Interconnect_module_component (see 4.2.442) where mounting restrictions for components exist. The shape representing the Mounting_restriction_region shall have an origin that is congruent with the origin of the shape representing the Interconnect_module_component. This congruency shall be accomplished using concepts equivalent to the Application objects Component_2d_position or Component_3d_position. The shape representing the Mounting_restriction_area is itself in the shape of the assembly, as is the shape of the Interconnect_module_component. The congruency requirement is established so that an application interface can more unambiguously import the Mounting_restriction_area into a CAD layout tool. Mounting_restriction_area is part of a design requirement for an interconnect that is established in the assembly or assemblies in which the interconnect is used. The interconnect shall specify the assembly or assemblies it is designed to support by the assembly_design_requirement attribute of Interconnect_module (see 4.2.440) if the attribute is present.

The data associated with a Mounting_restriction_area are the following:

- mounting_surface;
- scope.

4.2.550.1 mounting_surface

The mounting_surface specifies the role of the Interconnect_module_component_surface_feature (see 4.2.444) for the Mounting_restriction_area. There shall be one surface in the role of mounting_surface for a Mounting_restriction_area. The type of surface is determined by the definition of the Interconnect_module_component_surface_feature specified by mounting_surface. The mounting_surface conveys design intent to identify the component that implements the connectivity prescribed by the relevant member of Physical_connectivity_definition (see 4.2.652).

4.2.550.2 scope

The scope specifies the role of the Assembly_module (see 4.2.43) for the Mounting_restriction_area. The element_shape is within the Assembly_module geometric model coordinate system.

Constraints:

WR1: The component associated with the mounting_surface shall be a component in the scope.

NOTE EXPRESS specification:

```

*)
ENTITY mounting_restriction_area
  SUBTYPE OF (non_feature_shape_element);
    mounting_surface : interconnect_module_component_surface_feature;
    SELF\non_feature_shape_element.scope : assembly_module;
WHERE
  WR1: 'AP210_ARM.CSG_2D_SHAPE' IN
    TYPEOF(SELF\non_feature_shape_element.element_shape);
END_ENTITY; -- mounting_restriction_area
( *
```

4.2.551 Mounting_restriction_volume

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Mounting_restriction_volume` is a type of `Non_feature_shape_element` (see 4.2.560). A `Mounting_restriction_volume` is the region on an `Interconnect_module_component` (see 4.2.442) where mounting restrictions for components exist. The shape representing the `Mounting_restriction_region` shall have an origin that is congruent with the origin of the shape representing the `Interconnect_module_component`. This congruency shall be accomplished using concepts equivalent to the Application objects `Component_2d_position` or `Component_3d_position`. The shape representing the `Mounting_restriction_volume` is itself in the shape of the assembly, as is the shape of the `Interconnect_module_component`. The congruency requirement is established so that an application interface can more unambiguously import the `Mounting_restriction_volume` into a CAD layout tool. `Mounting_restriction_volume` is part of a design requirement for an interconnect that is established in the assembly or assemblies in which the interconnect is used. The interconnect shall specify the assembly or assemblies it is designed to support by the `assembly_design_requirement` attribute of `Interconnect_module` (see 4.2.440) if the attribute is present.

The data associated with a `Mounting_restriction_volume` are the following:

- `volume`;
- `mounting_surface`;
- `scope`.

4.2.551.1 **volume**

The `volume` is an inverse attribute that specifies the `Shape_definition` the `Mounting_restriction_volume` is dependent on for existence.

4.2.551.2 **mounting_surface**

The `mounting_surface` specifies the role of the `Interconnect_module_component_surface_feature` (see 4.2.444) for the `Mounting_restriction_volume`. There shall be one surface in the role of `mounting_surface` for a `Mounting_restriction_volume`. The type of surface is determined by the definition of the `Interconnect_module_component_surface_feature` specified by `mounting_surface`. The `mounting_surface` conveys design intent to identify the component that implements the connectivity prescribed by the relevant member of `Physical_connectivity_definition` (see 4.2.652).

4.2.551.3 **scope**

The `scope` specifies the role of the `Assembly_module` (see 4.2.43) for the `Mounting_restriction_volume`. The `volume` is within the `Assembly_module` geometric model coordinate system.

Constraints:

WR1: The component associated with the `mounting_surface` shall be a component in the `scope`.

NOTE EXPRESS specification:

```

*)
ENTITY mounting_restriction_volume
  SUBTYPE OF (non_feature_shape_element);
    mounting_surface : interconnect_module_component_surface_feature;
    SELF\non_feature_shape_element.scope : assembly_module;
INVERSE
  volume : non_feature_shape_definition for associated_element;
WHERE
  WR1: 'AP210_ARM.BOUND_VOLUME_SHAPE' IN TYPEOF(volume);
( *
  WR1: dimensionality (volume) = 3;
*)
END_ENTITY; -- mounting_restriction_volume

( *

```

4.2.552 Multi_layer_material_removal_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Multi_layer_material_removal_laminate_component is a type of Material_removal_laminate_component (see 4.2.535) that indicates that material is to be removed in multiple layers based on the same explicit geometry. The geometric transformations between the Stratum are used to position the geometry on the dependent Stratum.

NOTE This Application Object is not intended to be used in place of Cutout.

The data associated with a Multi_layer_material_removal_laminate_component are the following:

— derived_geometry.

4.2.552.1 derived_geometry

The derived_geometry specifies the role of the Layer (see 4.2.498) for the Multi_layer_material_removal_laminate_component. Each Stratum referenced by Layer receives the same material shape to within process tolerances.

NOTE EXPRESS specification:

```
*)
ENTITY multi_layer_material_removal_laminate_component
  SUBTYPE OF (material_removal_laminate_component);
  derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_material_removal_laminate_component

( *
```

4.2.553 Multi_layer_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Multi_layer_stratum_feature is a type of Stratum_feature (see 4.2.851). A Multi_layer_stratum_feature causes shapes to be realized in the material of multiple sequential Stratum (see 4.2.845) which are all bases on the geometry for the feature. Each shape is effectively a Stratum_feature in that Stratum, but due to the identical nature of the geometry, there is no need to separately model the geometry. Each shape in the sequential Stratum is isolated from other shapes in that Stratum.

EXAMPLE 1 Gold plating on Conductors that is based directly on the underlying Copper shape is an example of this Application Object.

The data associated with a Multi_layer_stratum_feature are the following:

— derived_layer_geometry.

4.2.553.1 derived_layer_geometry

The derived_layer_geometry specifies the role of the Stratum (see 4.2.845) for the Multi_layer_stratum_feature.

NOTE EXPRESS specification:

```
*)
ENTITY multi_layer_stratum_feature
  SUBTYPE OF (stratum_feature);
  derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_stratum_feature

( *
```

4.2.554 Multi_layer_stratum_feature_template_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Multi_layer_stratum_feature_template_component` is a type of `Stratum_feature_template_component` (see 4.2.862). A `Multi_layer_stratum_feature_template_component` is included to allow the creation of shapes on sequential `Stratum` where those shapes are identical to the shape that references the `Multi_layer_stratum_feature_template_component`.

EXAMPLE 1 The domain feature known as "edge finger connector" consists of multiple `Stratum` of Copper, Nickel, Gold and each material shape is based on the shape of the underlying `Stratum_feature_template_component` in the Copper `Stratum`.

The data associated with a `Multi_layer_stratum_feature_template_component` are the following:

— `derived_layer_geometry`.

4.2.554.1 `derived_layer_geometry`

The `derived_layer_geometry` specifies the role of the `Layer` (see 4.2.498) for the `Multi_layer_stratum_feature_template_component`.

NOTE EXPRESS specification:

```
*)
ENTITY multi_layer_stratum_feature_template_component
  SUBTYPE OF (stratum_feature_template_component);
    derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_stratum_feature_template_component

( *
```

4.2.555 `Next_higher_assembly_interface_requirement`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Next_higher_assembly_interface_requirement` is a type of `Managed_design_object` (see 4.2.524). A `Next_higher_assembly_interface_requirement` is the association between the version of the next higher assembly and a shape that simulates the fit characteristics of the next higher assembly. The coordinate system of the shape is congruent with the coordinate system of the current design. The shape simulates a maximum material condition state of the next higher assembly.

The data associated with a `Next_higher_assembly_interface_requirement` are the following:

— `shape`;

— `version_of_next_higher_assembly`.

4.2.555.1 shape

The shape specifies the Bound_volume_shape (see 4.2.76) for the model for the Next_higher_assembly_interface_requirement.

4.2.555.2 version_of_next_higher_assembly

The version_of_next_higher_assembly specifies the Ee_product_version (see 4.2.328) of the Next_higher_assembly_interface_requirement.

NOTE EXPRESS specification:

```

*)
ENTITY next_higher_assembly_interface_requirement
  SUBTYPE OF (managed_design_object);
    shape : bound_volume_shape;
    version_of_next_higher_assembly : ee_product_version;
END_ENTITY; -- next_higher_assembly_interface_requirement

( *
```

4.2.556 Next_higher_assembly_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Next_higher_assembly_relationship is a type of Assembly_composition_relationship (see 4.2.39). A Next_higher_assembly_relationship is the association of a component to the assembly it helps to compose.

EXAMPLE 1 Figure 63 illustrates roles of several components in the related assembly.

The data associated with a Next_higher_assembly_relationship are the following:

- assembly_position_2d;
- assembly_position_3d;
- quantity;
- reference_designation.

4.2.556.1 assembly_position_2d

The assembly_position_2d is an inverse relationship that specifies the Component_assembly_2d_position (see 4.2.112) that specifies the Next_higher_assembly_relationship as its component. There shall be no more than one Component_assembly_2d_position for a Next_higher_assembly_relationship.

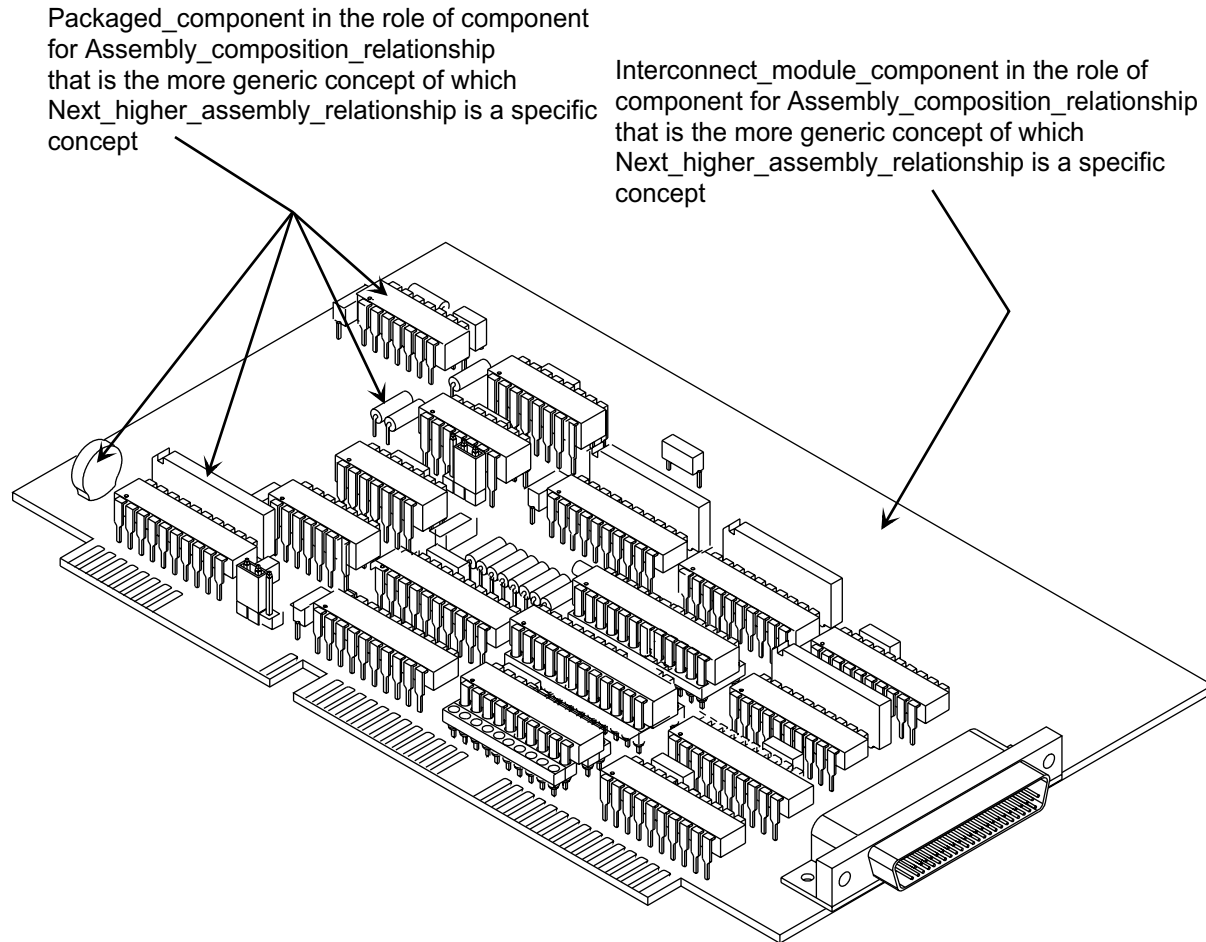


Figure 63 – Next_higher_assembly_relationship

4.2.556.2 assembly_position_3d

The assembly_position_3d is an inverse relationship that specifies the Component_assembly_3d_position (see 4.2.113) that specifies the Next_higher_assembly_relationship as its component. There shall be no more than one Component_assembly_3d_position for a Next_higher_assembly_relationship.

4.2.556.3 quantity

The quantity specifies the role of the Data_element (see 4.2.211) for the Next_higher_assembly_relationship. The quantity need not be specified for a particular Next_higher_assembly_relationship.

4.2.556.4 reference_designation

The reference_designation specifies the ee_name as an identifier for the Next_higher_assembly_relationship.

NOTE Some components, such as mechanical fasteners, may not be identified by reference_designations within a PCA. They are tracked in the Bill of Material for the PCA by part number and quantity. This application practice varies among enterprises. This practice is supported by Assembly_composition_relationship and not by Next_higher_assembly_relationship.

Constraints:

UR1: The combination of reference_designation and the assembly attribute inherited from the supertype Assembly_composition_relationship shall be unique within a population of Next_higher_assembly_relationship.

IP1: The quantity shall be provided only when the related Assembly_component is an Interconnect_component, and the quantity shall be a length unit.

IP2: Things shall not be located on themselves.

NOTE EXPRESS specification:

```

*)
ENTITY next_higher_assembly_relationship
  SUBTYPE OF (assembly_composition_relationship);
  --tt tc1 wd10 modified
  reference_designation : OPTIONAL ee_name;
  quantity               : OPTIONAL data_element;
  INVERSE
    assembly_position_3d : SET [0:1] OF
                          component_assembly_3d_position FOR component;
    assembly_position_2d : SET [0:1] OF
                          component_assembly_2d_position FOR component;
  UNIQUE
    UR1 : reference_designation, assembly;
  WHERE
    WR1: (EXISTS(reference_designation)) XOR
          (EXISTS(quantity));
END_ENTITY; -- next_higher_assembly_relationship

( *
```

4.2.557 Non_conductive_base_blind_via

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Non_conductive_base_blind_via` is a type of `Blind_via` (see 4.2.71) with a floor whose material is non-conductive.

NOTE EXPRESS specification:

```
*)
ENTITY non_conductive_base_blind_via
  SUBTYPE OF (blind_via);
END_ENTITY; -- non_conductive_base_blind_via

--tt tc1 correction, printed_part_template not implementable
(*
```

4.2.558 `Non_conductive_cross_section_template`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Non_conductive_cross_section_template` is a type of a `Part_template` (see 4.2.620).

The data associated with a `Non_conductive_cross_section_template` are the following:

- `template_technology`;
- `nominal_width`.

4.2.558.1 `template_technology`

The `template_technology` specifies a `Stratum_technology` (see 4.2.871) for the `Non_conductive_cross_section_template`.

4.2.558.2 `nominal_width`

The `nominal_width` specifies a `Length_data_element` (see 4.2.511) for the `Non_conductive_cross_section_template`.

NOTE EXPRESS specification:

```
*)
ENTITY non_conductive_cross_section_template
  SUBTYPE OF (part_template);
  template_technology : stratum_technology;
  nominal_width : length_data_element;
END_ENTITY; -- non_conductive_cross_section_template

(*
```

4.2.559 Non_feature_shape_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Non_feature_shape_definition is a type of Shape_definition (see 4.2.808). A Non_feature_shape_definition may be one of a View_level_non_feature_shape_definition (see 4.2.955), or a Pre_defined_datum_axis_3d_symbol (see 4.2.705), or a Pre_defined_datum_plane_3d_symbol (see 4.2.706), or a Pre_defined_parallel_datum_axis_2d_symbol (see 4.2.709), or a Pre_defined_perpendicular_datum_axis_2d_symbol (see 4.2.711), or a Pre_defined_perpendicular_datum_plane_2d_symbol (see 4.2.713). A Non_feature_shape_definition may also be one of a Physical_unit_3d_keepout_shape (see 4.2.665), or a Physical_unit_planar_keepout_shape (see 4.2.669). A Non_feature_shape_definition shall be used to represent annotation information or other shape property information that is not appropriate for Feature_shape_definition.

NOTE Shape information that is derived from a physical shape and whose algorithmic derivation details are unavailable may be exchanged with this Application Object.

The data associated with a Non_feature_shape_definition are the following:

- associated_element;
- model_shape.

4.2.559.1 associated_element

The associated_element specifies the role of the Non_feature_shape_element (see 4.2.560) for the Non_feature_shape_definition.

4.2.559.2 model_shape

The model_shape specifies the role of the Feature_shape_definition (see 4.2.372) for the Non_feature_shape_definition. The model_shape need not be supplied.

NOTE EXPRESS specification:

```

*)
ENTITY non_feature_shape_definition
  SUPERTYPE OF ((ONEOF(
    pre_defined_datum_axis_3d_symbol,
    pre_defined_datum_plane_3d_symbol,
    pre_defined_parallel_datum_axis_2d_symbol,
    pre_defined_perpendicular_datum_axis_2d_symbol,
    pre_defined_perpendicular_datum_plane_2d_symbol,
    view_level_non_feature_shape_definition)) ANDOR
    (ONEOF(physical_unit_3d_keepout_shape,
    physical_unit_planar_keepout_shape)));
  SUBTYPE OF (shape_definition);
    associated_element : non_feature_shape_element;
    model_shape       : OPTIONAL feature_shape_definition;
END_ENTITY; -- non_feature_shape_definition
( *

```

4.2.560 Non_feature_shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Non_feature_shape_element is a type of Shape_element (see 4.2.810) that is not on the physical boundary of a Physical_unit (see 4.2.664). A Non_feature_shape_element may be one of a Datum_-reference_frame (see 4.2.235), a Datum_system (see 4.2.236), a Derived_shape_element (see 4.2.260), a Directed_axis (see 4.2.303), a Positional_boundary (see 4.2.698), a Positional_boundary_member (see 4.2.701), a Tolerance_zone (see 4.2.913), a Tolerance_zone_boundary (see 4.2.915), a Tolerance_zone_-boundary_member (see 4.2.917), an Interconnect_module_constraint_region (see 4.2.446), a Mounting_restriction_area (see 4.2.550), a Mounting_restriction_volume (see 4.2.551), or a Viewing_plane (see 4.2.956). Additionally, a Non_feature_shape_element may be a Datum (see 4.2.214).

The data associated with a Non_feature_shape_element are the following:

- element_shape;
- scope.

EXAMPLE 1 The centre axis of a cylindrical hole is a Non_feature_shape_element.

4.2.560.1 element_shape

The element_shape may specify the role of a Feature_shape_occurrence (see ??), for the Non_feature_shape_element.

4.2.560.2 scope

The scope may specify the role of a Footprint_definition (see 4.2.388), a Padstack_definition (see 4.2.595), a Physical_unit (see 4.2.664), a Stratum (see 4.2.845), or a Template_definition (see 4.2.887) for the Non_feature_shape_element.

Constraints:

UR1: The combination of name inherited from Shape_element and scope shall be unique.

WR1: If the Non_feature_shape_element is a Datum it shall not also be a Datum_reference_frame, a Datum_system, a Directed_axis, a Positional_boundary, a Positional_boundary_member, a Tolerance_zone, a Tolerance_zone_boundary, a Tolerance_zone_boundary_member, or a Viewing_plane.

IP1: The Non_feature_shape_element shall not be on the boundary of a Physical_unit.

NOTE EXPRESS specification:

```

*)
ENTITY non_feature_shape_element
  SUPERTYPE OF (ONEOF (datum_reference_frame, datum_system,
    derived_shape_element, directed_axis, positional_boundary,
    positional_boundary_member, tolerance_zone, tolerance_zone_boundary,
    tolerance_zone_boundary_member, viewing_plane,
    seating_plane, template_material_cross_section_boundary,
    interconnect_module_constraint_region,
    mounting_restriction_area,
    mounting_restriction_volume
  ) ANDOR datum)
  SUBTYPE OF (shape_element);
  scope : ee_product_definition_with_annotation_elements;
INVERSE
  element_shape : SET [0:?] OF non_feature_shape_definition FOR
    associated_element;
  associated_usage_concept : SET [0:?] OF usage_concept_usage_relationship FOR
    associated_usage;
UNIQUE
  UR1: SELF\shape_element.name, scope;
WHERE
  WR1: (NOT (SIZEOF(associated_usage_concept) &#62; 0 ) ) OR
    (SIZEOF(element_shape) &#62; 0);
( *
physical_unit_design_view in
TYPEOF element_shape.associated_composition_relationship.
  resultant_feature_shape\physical_unit_3d_shape.shape_characterized_physical_u
OR
physical_unit_design_view in
TYPEOF element_shape.associated_composition_relationship.
  resultant_feature_shape\physical_unit_planar_shape.shape_characterized_physic
*)
( *
WR2: SIZEOF(QUERY(es &#60;* element_shape | NOT
  (('AP210_ARM.PHYSICAL_UNIT_DESIGN_VIEW' IN TYPEOF(
    es.associated_composition_relationship.
    resultant_feature_shape\physical_unit_3d_shape.
    shape_characterized_physical_unit))) OR
  ('AP210_ARM.PHYSICAL_UNIT_DESIGN_VIEW' IN TYPEOF(
    es.associated_composition_relationship.
    resultant_feature_shape\physical_unit_planar_shape.
    shape_characterized_physical_unit))))
  )) = 0;
*)
END_ENTITY; -- non_feature_shape_element
( *

```

4.2.561 Non_feature_shape_element_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Non_feature_shape_element_relationship` is a type of `Shape_element_relationship` (see 4.2.816) that associates a `Derived_shape_element` (see 4.2.260) with a `Shape_element` (see 4.2.810) from which the `Derived_shape_element` is derived.

The data associated with a `Non_feature_shape_element_relationship` are the following:

- `related_shape_element`;
- `relating_shape_element`.

4.2.561.1 `related_shape_element`

The `related_shape_element` specifies the `Non_feature_shape_element` (see 4.2.560) for the `Non_feature_shape_element_relationship`.

4.2.561.2 `relating_shape_element`

The `relating_shape_element` specifies the `Non_feature_shape_element` (see 4.2.560) for the `Non_feature_shape_element_relationship`.

NOTE The attribute `related_shape_element` is declared in the `Shape_element_relationship` entity of which this entity is a subtype.

Constraints:

UR1: The combination of the `relating_shape_element` inherited from `Shape_element_relationship` and `related_shape_element` inherited from `Shape_element_relationship` shall be unique within a population of `Non_feature_shape_element_relationship`.

NOTE EXPRESS specification:

```

*)
ENTITY non_feature_shape_element_relationship
  SUBTYPE OF (shape_element_relationship);
    SELF\shape_element_relationship.related_shape_element : non_feature_shape_element;
    SELF\shape_element_relationship.relying_shape_element : non_feature_shape_element;
  UNIQUE
    UR1 : related_shape_element, relating_shape_element;
END_ENTITY; -- non_feature_shape_element_relationship

( *
```

4.2.562 `Non_functional_land`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Non_functional_land` is a type of `Stratum_feature_template_component` (see 4.2.862) whose implementing `Stratum_feature` (see 4.2.851) is not also being composed by other instances of `Laminate_component` (see 4.2.483). A `Non_functional_land` may be one of a `Component_termination_passage_and_contact_size_dependent_non_functional_land` (see 4.2.145), a `Component_termination_passage_dependent_non_functional_land` (see 4.2.147), a `Contact_size_dependent_non_functional_land` (see 4.2.195), an `Unsupported_passage_dependent_non_functional_land` (see 4.2.940), a `Via_and_contact_size_dependent_non_functional_land` (see 4.2.949), or a `Via_dependent_non_functional_land` (see 4.2.951).

The data associated with a `Non_functional_land` are the following:

- `access_mechanism`;
- `alternate_land_definition`;
- `usage_definition`.

4.2.562.1 `access_mechanism`

The `access_mechanism` is an inverse relationship that specifies that the existence of the `Non_functional_land` is dependent on the existence of the `Non_functional_land_join_terminal` (see 4.2.564) that specifies the `Non_functional_land` as its `associated_component`. There shall be one or more `Non_functional_land_join_terminal` for a `Non_functional_land`. The `access_mechanism` is only provided for support of plated passages within the lateral geometric boundary of the `Non_functional_land` and.

4.2.562.2 `alternate_land_definition`

The `alternate_land_definition` specifies one role of the `Land_physical_template` (see 4.2.493) for the `Non_functional_land`. The `alternate_land_definition` need not be specified for a particular `Non_functional_land`. An `alternate_land_definition` overrides the `usage_definition` if it is provided.

4.2.562.3 `usage_definition`

The `usage_definition` specifies one role of the `Land_physical_template` (see 4.2.493) for the `Non_functional_land`.

Constraints:

WR1: There shall be no direct connection to a `Conductive_interconnect_element` from a member of `Non_functional_land`.

WR2: The `stratum_feature` specified by `SELF\stratum_feature_template_component.stratum_feature_implementation` shall not be in the role of `stratum_feature_implementation` for any other `stratum_feature_template_component`.

NOTE EXPRESS specification:

```

*)
ENTITY non_functional_land
  SUPERTYPE OF (ONEOF (via_dependent_non_functional_land,
    component_termination_passage_dependent_non_functional_land,
    contact_size_dependent_non_functional_land,
    component_termination_passage_and_contact_size_dependent_non_functional_land,
    unsupported_passage_dependent_non_functional_land,
    via_and_contact_size_dependent_non_functional_land))
  SUBTYPE OF (stratum_feature_template_component);
  alternate_land_definition          : OPTIONAL land_physical_template;
  SELF\assembly_component.usage_definition : land_physical_template;
  INVERSE
    access_mechanism                : SET [1:?] OF
                                      non_functional_land_join_terminal FOR
                                      associated_component;
END_ENTITY; -- non_functional_land

( *
```

4.2.563 Non_functional_land_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Non_functional_land_interface_terminal is a type of Component_feature (see 4.2.118) included to identify the interface available for attachment in the next level of assembly.

NOTE Use of this capability occurs when a member of Component_feature_to_physical_usage_view_assignment (see 4.2.123) references a member of Non_functional_land_interface_terminal to assign a relationship to an Interconnect_module_terminal (see 4.2.463). A Non_functional_land_interface_terminal shall only be associated to types of Non_functional_land established for the purpose of interface attachment.

The data associated with a Non_functional_land_interface_terminal are the following:

- associated_component;
- definition.

4.2.563.1 associated_component

The associated_component specifies the role of the Non_functional_land (see 4.2.562) for the Non_functional_land_interface_terminal.

4.2.563.2 definition

The definition specifies the role of the Land_template_interface_terminal (see 4.2.494) for the Non_functional_land_interface_terminal.

Constraints:

WR1: The Land_template that is the usage_definition for the Non_functional_land in the role of associated_component for this Non_functional_land_interface_terminal is the associated_definition for the definition of this Non_functional_land_interface_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY non_functional_land_interface_terminal
  SUBTYPE OF (laminare_component_interface_terminal);
    SELF\component_feature.definition      : land_template_interface_terminal;
    SELF\component_feature.associated_component : non_functional_land;
WHERE
  WR1: SIZEOF([ 'AP210_ARM.CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND',
                'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_NON_FUN
                'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND' ] *
            TYPEOF(SELF\component_feature.associated_component))
    = 1;
END_ENTITY; -- non_functional_land_interface_terminal

( *
```

4.2.564 Non_functional_land_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Non_functional_land_join_terminal is a type of Component_terminal (see 4.2.137) that is included in a design to support the description of the relationship between the Non_functional_land and a plated passage that implements a connection to a feature on a different member of Stratum (see 4.2.845) than the member of Stratum the Non_functional_land_join_terminal is associated with.

The data associated with a Non_functional_land_join_terminal are the following:

- associated_component;
- definition.

4.2.564.1 associated_component

The associated_component specifies the role of the Non_functional_land (see 4.2.562) for the Non_functional_land_join_terminal.

4.2.564.2 definition

The definition specifies the role of the Land_template_join_terminal (see 4.2.495) for the Non_functional_land_join_terminal. The valid purposes for a Non_functional_land_join_terminal are to connect to a Via_terminal or a Component_termination_passage_join_terminal.

Constraints:

WR1: The Land_template that is the usage_definition for the Non_functional_land this Non_functional_land_join_terminal is a terminal of is the associated_definition for the definition of this Non_functional_land_join_terminal.

WR2: A Non_functional_land_join_terminal is only to be connected to a passage join terminal, and to nothing else. It is not allowed to be connected to a trace on the same layer; the join is required to be to a passage within the lateral extent of the non-functional land.

NOTE EXPRESS specification:

```
*)
ENTITY non_functional_land_join_terminal
  SUBTYPE OF (laminar_component_join_terminal);
    SELF\component_feature.definition      : land_template_join_terminal;
    SELF\component_feature.associated_component : non_functional_land;
END_ENTITY; -- non_functional_land_join_terminal

( *
```

4.2.565 Numeric_parameter

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Numeric_parameter is a type of Characteristic (see 4.2.98). A Numeric_parameter is a quantitative value in an identified unit of measure. Uncertainty may be applied to the value by referencing the Qualified_data_element (see ??) type of Data_element.

NOTE Actual values of occurrence data will be dependent on many variables.

The data associated with a Numeric_parameter are the following:

— property_value.

4.2.565.1 property_value

The property_value specifies the Data_element (see 4.2.211) for the quantitative value and unit of measure for the Numeric_parameter.

NOTE EXPRESS specification:

```

*)
ENTITY numeric_parameter
  SUBTYPE OF (characteristic);
  SELF\characteristic.property_value : data_element;
END_ENTITY; -- numeric_parameter

--tt tc1 wd9
( *
```

4.2.566 Open_shell_based_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Open_shell_based_surface` is a type of `Feature_shape_definition` (see 4.2.372) that represents a shape with a simple or complex surface using a set of connected faces. The shape shall be manifold and shall not be closed. The shape shall not be used in the composition of a closed volume. The shape shall be a single shell if it is the shape of a `Stratum_feature` (see 4.2.851), `Stratum_feature_template_component` (see 4.2.862), `Stratum_feature_template` (see 4.2.861), or the shape of a `Design_intent_modification` (see 4.2.267). The shape shall be one or more shell if it is the shape of a `Stratum` (see 4.2.845). The shape shall be a single shell if it is the shape of a `Stratum_average_surface` (see 4.2.847) or the shape of a `Stratum_surface` (see 4.2.868). An `Open_shell_based_surface` guarantees material continuity as the material traverses corners in a three-dimensional model.

NOTE EXPRESS specification:

```

*)
ENTITY open_shell_based_surface
  SUPERTYPE OF (ONEOF(
    stratum_average_surface_shape,
    stratum_surface_shape,
    design_intent_modification_non_planar_2d_shape,
    part_template_non_planar_2d_shape,
    stratum_feature_non_planar_2d_shape))
  SUBTYPE OF (shape_definition);
END_ENTITY; -- open_shell_based_surface

( *
```

4.2.567 Operational_requirement_occurrence_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Operational_requirement_occurrence_relationship` is a type of `Managed_design_object` (see 4.2.524). An `Operational_requirement_occurrence_relationship` is the association between one `Ee_requirement_occurrence` (see 4.2.330) and another, with a defined type of relationship. An `Operational_requirement_occurrence_relationship` is an incompletely modeled concept and shall be combined with members of `Requirement_composition` (see 4.2.770) to form a complete requirement.

The data associated with an `Operational_requirement_occurrence_relationship` are the following:

- `related_requirement_occurrence`;
- `relating_requirement_occurrence`;
- `relationship_type`.

4.2.567.1 `related_requirement_occurrence`

The `related_requirement_occurrence` specifies the first `Ee_requirement_occurrence` (see 4.2.330) for the `Operational_requirement_occurrence_relationship`. The first `Ee_requirement_occurrence` is the dependent item if the relationship is a dependency relationship.

4.2.567.2 `relating_requirement_occurrence`

The `relating_requirement_occurrence` specifies the second `Ee_requirement_occurrence` (see 4.2.330) for the `Operational_requirement_occurrence_relationship`. The second `Ee_requirement_occurrence` is the independent item if the relationship is a dependency relationship.

4.2.567.3 `relationship_type`

The `relationship_type` specifies one of the following:

- an `AND_OPERATION`;
- an `OR_OPERATION`;
- a `DERIVED_FROM` relationship.

The mathematical operations `AND`, `OR` shall be implemented for this `relationship_type`. `DERIVED_FROM` only asserts a dependency relationship. The `AND_OPERATION` is provided so that changes in `relationship_type` from `OR_OPERATION` to `AND_OPERATION` do not cause a structural change to the populated data. The implicit relationship between two member of `Ee_requirement_occurrence` that help to compose the same `Requirement_group_occurrence` is an `AND` relationship.

Constraints:

WR1: Each member of `Operational_requirement_occurrence_relationship` shall specify members of `Ee_requirement_occurrence` that are also specified by a member of `Requirement_composition` that specifies a member of `Requirement_group_occurrence` as the composition or result of the operation.

NOTE EXPRESS specification:

```
*)
ENTITY operational_requirement_occurrence_relationship
  SUBTYPE OF (managed_design_object);
    relationship_type      : relationship_operation;
    relating_requirement_occurrence : ee_requirement_occurrence;
    related_requirement_occurrence  : ee_requirement_occurrence;
END_ENTITY; -- operational_requirement_occurrence_relationship

( *
```

4.2.568 `Opposing_boundary_set_size_characteristic`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An `Opposing_boundary_set_size_characteristic` is a type of `Boundary_size_characteristic` (see 4.2.78) that defines the distance between the two members of a `Tolerance_zone_opposing_boundary_set` (see 4.2.926). The two members may or may not be explicitly identified.

The data associated with an `Opposing_boundary_set_size_characteristic` are the following:

— `sized_boundary_set`.

4.2.568.1 `sized_boundary_set`

The `sized_boundary_set` specifies the `Tolerance_zone_opposing_boundary_set` (see 4.2.926) of which the distance between its two members is defined by the `Opposing_boundary_set_size_characteristic`.

Constraints:

WR1: The `Tolerance_zone_boundary` specified as the `sized_boundary_set` shall be the same instance of `Tolerance_zone_boundary` as specified by the `defined_zone_boundary` of the `Boundary_zone_definition_with_specified_size` that is specified as the `boundary_definition`.

NOTE The attribute `boundary_definition` is declared in the `Boundary_size_characteristic` entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY opposing_boundary_set_size_characteristic
  SUBTYPE OF (boundary_size_characteristic);
    sized_boundary_set : tolerance_zone_opposing_boundary_set;
END_ENTITY; -- opposing_boundary_set_size_characteristic

( *

```

4.2.569 Organization

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Organization is a type of Managed_design_object (see 4.2.524). An Organization is a collection of persons grouped for a specific purpose.

The data associated with an Organization are the following:

- address;
- identifier.

4.2.569.1 address

The address specifies the string that describes the location where the Organization may be reached by postal mail or in person. The address need not be specified for a particular Organization.

4.2.569.2 identifier

The identifier specifies the name of the Organization.

EXAMPLE 1 The identifier may be a concatenation of the following: "Country code" = country code (as established by UN); "Organization code name" = identifying name the country gives the code; "Organization code" = organization code (as established by country assigning the organization code); "Sub Organization code" = sub organization code (as established by organization). This illustrates the usage: "Country code" = "USA", "Organization code name" = "CAGE", "Organization code" = "12345", "Sub Organization code" = "Telecommunications".

NOTE Reference ISO 6523 amendment 1:1994.

Constraints:

UR1: The identifier shall be unique within a population of Organizations.

NOTE EXPRESS specification:

```

*)
ENTITY organization
  SUBTYPE OF (managed_design_object);
  identifier : ee_name;
  address    : OPTIONAL STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- organization

( *
```

4.2.570 Owned_design_object

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Owned_design_object is a type of Managed_design_object (see 4.2.524). An Owned_design_object identifies the Managed_design_object which participates in the Owned_assignment (see ??).

NOTE The purpose of Owned_design_object is identify the single authority ownership of design aspects. The typical use is to negotiate between the owners of the different design aspects. If there are multiple joint owners, then the Frozen_design_object is more appropriate since that uses Ee_approval which supports multiple signatures.

NOTE EXPRESS specification:

```

*)
ENTITY owned_design_object
  SUBTYPE OF (managed_design_object);
  INVERSE
    owner : owner_assignment for owned_item;
END_ENTITY; -- owned_design_object

( *
```

4.2.571 Owner_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Owner_assignment is a type of Managed_design_object (see 4.2.524). An Owner_assignment is the specification that one or more Owned_design_object (see 4.2.570) are controlled by a specific entity during design activity.

The data associated with an Owner_assignment are the following:

- assigned_owner;
- owned_item.

4.2.571.1 assigned_owner

An assigned_owner specifies the role of the Person_and_organization (see 4.2.647) for the Owner_assignment.

4.2.571.2 owned_item

The owned_item specifies an Owned_design_object (see 4.2.570) for which the design is owned by the assigned_owner. There shall be one or more Owned_design_object for a Owner_assignment.

Constraints:

WR1: Each Owner_assignment shall not be referenced by any Owner_assignment.

NOTE EXPRESS specification:

```
*)  
ENTITY owner_assignment  
  SUBTYPE OF (managed_design_object);  
    owned_item      : SET [1:?] OF owned_design_object;  
    assigned_owner : person_and_organization;  
END_ENTITY; -- owner_assignment  
  
(*
```

4.2.572 Movable_packaged_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Movable_packaged_component_join_terminal is a type of Packaged_component_join_terminal (see 4.2.586). A Movable_packaged_component_join_terminal is one termination for a Movable_packaged_component (see ??). The terminal provides access to the electrical functionality of the movable_packaged part being terminated.

The data associated with a Movable_packaged_component_join_terminal are the following:

- maximum_wire_terminal_length;
- minimum_wire_terminal_length.

4.2.572.1 maximum_wire_terminal_length

The maximum_wire_terminal_length specifies the maximum dimension of the Length_data_element (see 4.2.511) of the Movable_packaged_component_join_terminal.

4.2.572.2 minimum_wire_terminal_length

The minimum_wire_terminal_length specifies the minimum dimension of the Length_data_element (see 4.2.511) of the Movable_packaged_component_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY movable_packaged_component_join_terminal
  SUBTYPE OF (packaged_component_join_terminal)
    maximum_wire_terminal_length          : length_data_element;
    minimum_wire_terminal_length          : length_data_element;
WHERE
  WR1: 'AP210_ARM.WIRE_TERMINAL' IN
    TYPEOF(SELF\component_feature.definition\packaged_part_terminal.
      terminal_of_package);
END_ENTITY; -- movable_packaged_component_join_terminal
( *
```

4.2.573 Package

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package is a type of a Physical_unit_usage_view (see 4.2.672). A Package may be a Altered_package (see 4.2.8). A Package describes the physical container of a Packaged_part (see 4.2.591). A Package includes no more than one package body and one or more package terminals.

EXAMPLE 1 Figures 64 through 69 illustrate types of Packages and parameters related to Geometric Dimensioning and Tolerancing supported by this part of ISO 10303. ISO 10303-202 would be used in conjunction with this part of ISO 10303 to define the exchange of the presentation information (i.e., the drawings). This part of ISO 10303 supports the direct application to application exchange of the parametric data illustrated in the drawings.

EXAMPLE 2 Figures 87 through 93 illustrate parameters and features related to Package orientation for assembly purposes supported by this part of ISO 10303. ISO 10303-202 would be used in conjunction with this part of ISO 10303 to define the exchange of the presentation information (i.e., the drawings).

NOTE When used for a Packaged_part "white box model", the Part_device (see 4.2.606) is installed in the Package and an electrical connection is defined between each terminal on the installed Part_device and one or more terminals on the Package.

The data associated with a Package are the following:

- body;
- cutout_required;
- least_lead_length_below_seating_plane;
- least_material_condition_centroid_location;
- maximum_body_clearance_above_seating_plane;
- maximum_body_clearance_below_seating_plane;
- maximum_body_height_above_seating_plane;
- maximum_body_height_below_seating_plane;
- maximum_lead_length_below_seating_plane;
- maximum_material_condition_centroid_location;
- minimum_body_clearance_above_seating_plane;
- minimum_body_clearance_below_seating_plane;
- package_accesses;
- package_polarity_indication;
- package_primary_orientation_feature;
- interface_plane;
- package_seating_plane;
- package_secondary_orientation_feature;
- package_tertiary_orientation_feature;
- mounting_technology.

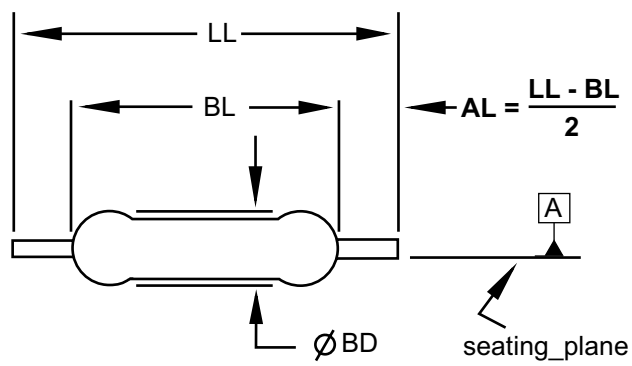


Figure 64 – Axial lead

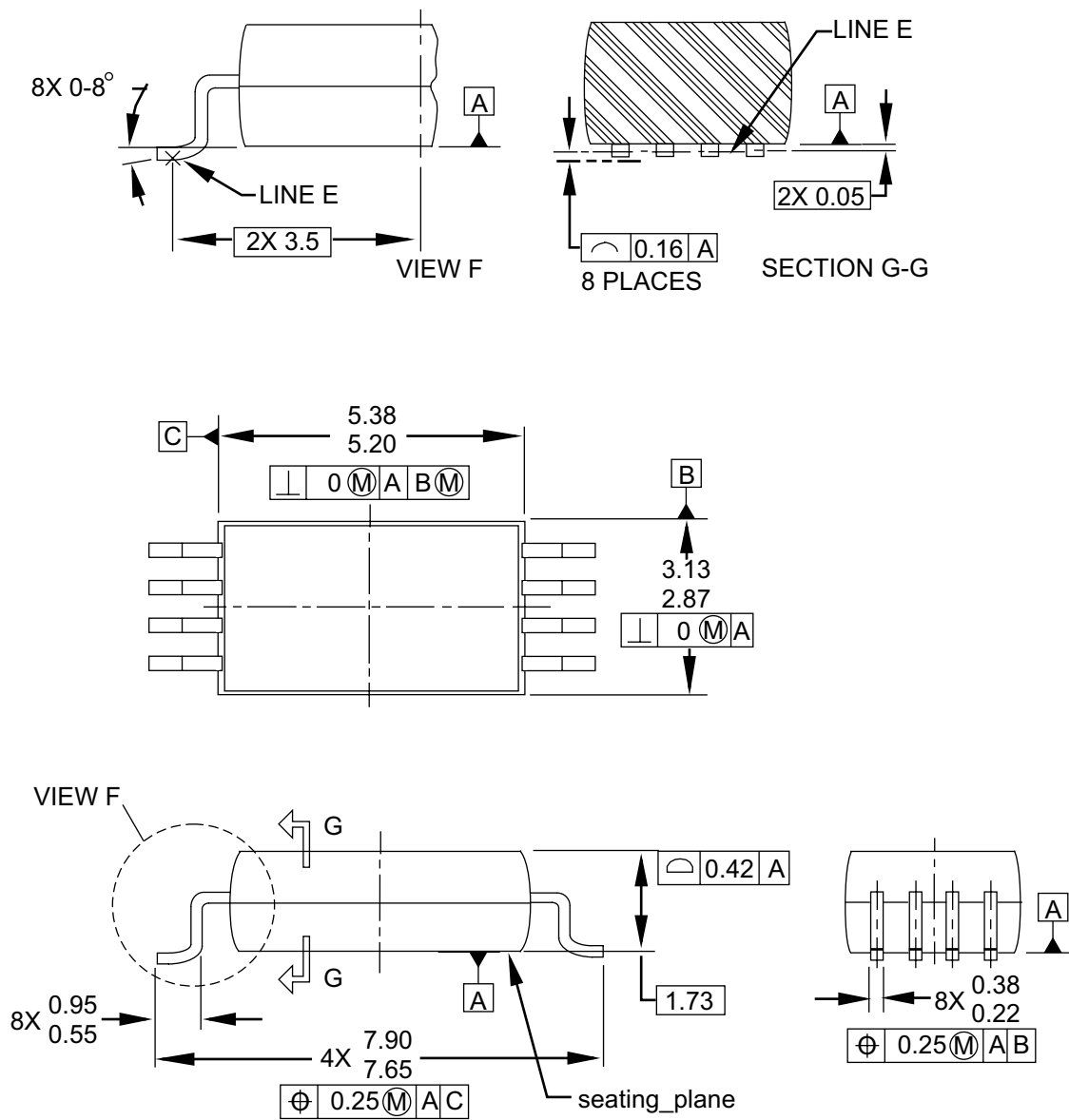


Figure 65 – Eight lead SOIC

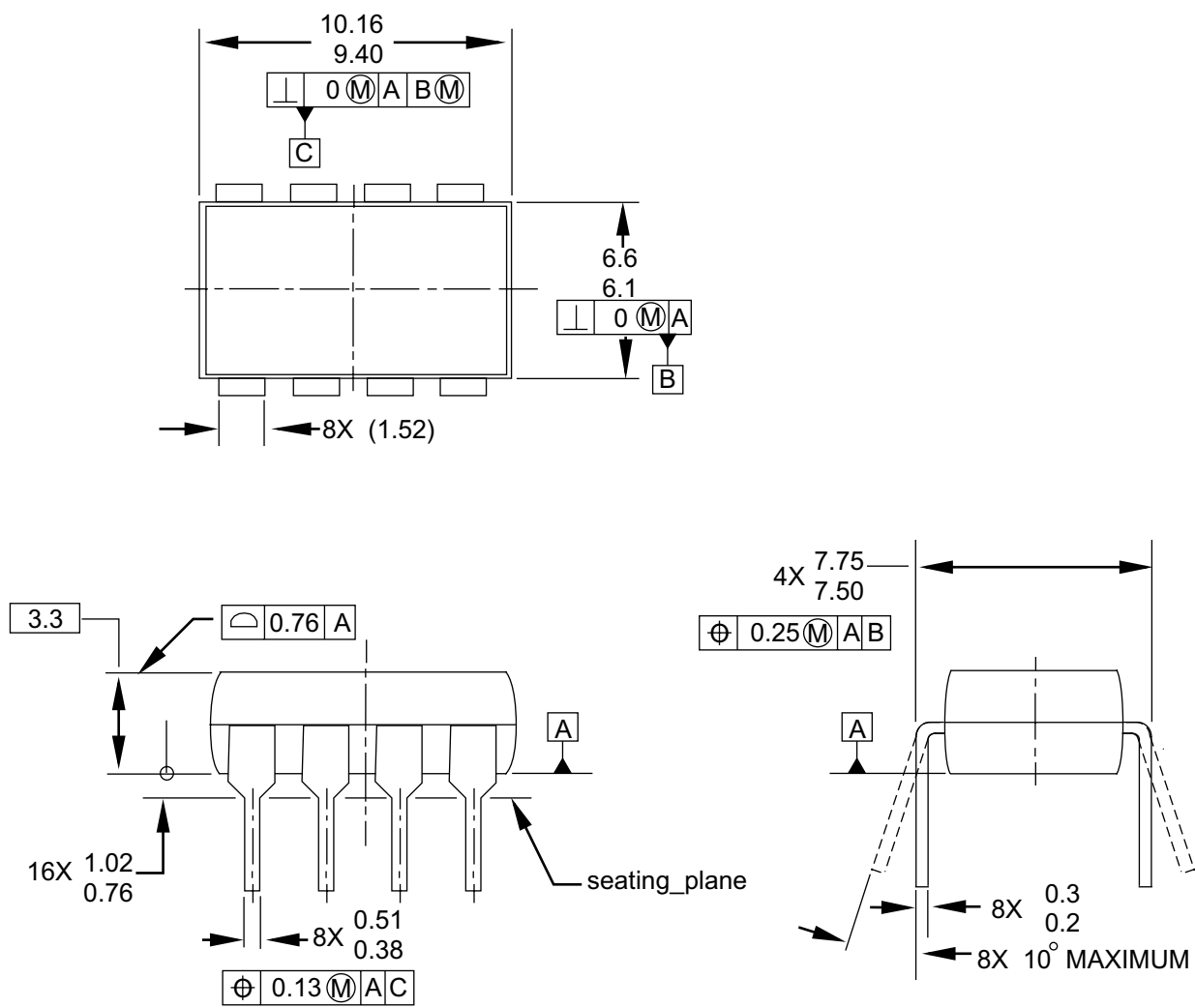
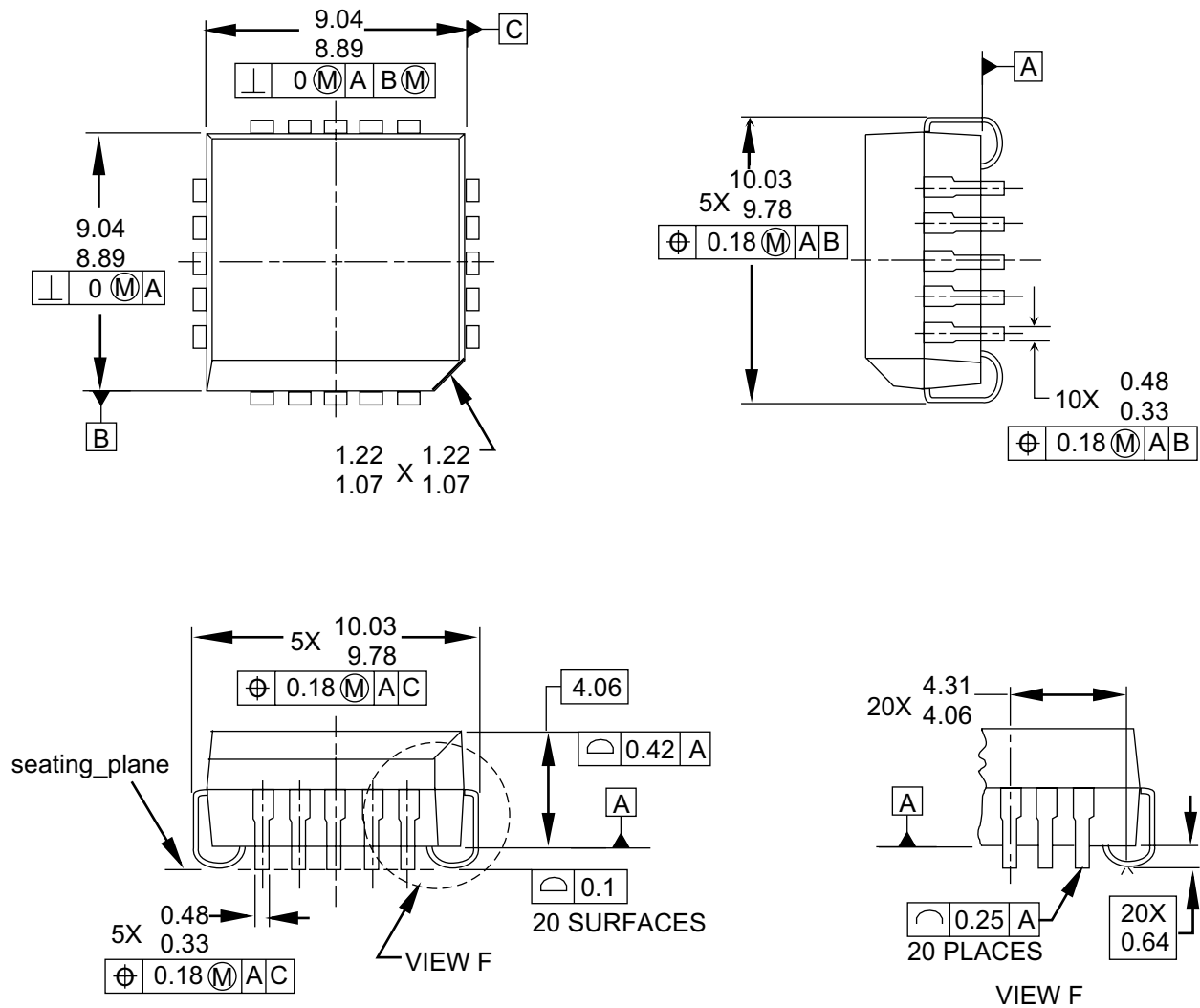


Figure 66 – Eight lead DIP



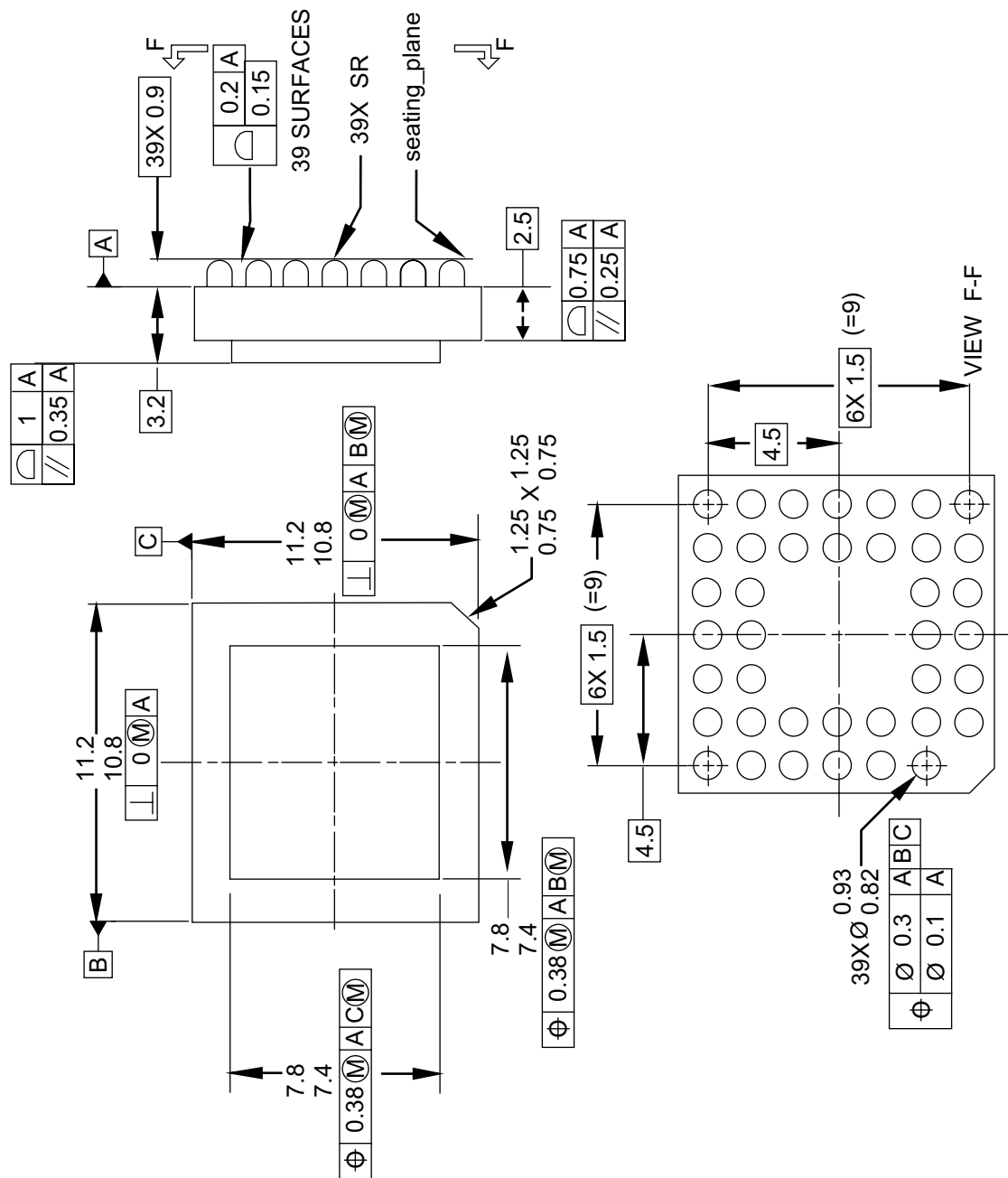


Figure 68 – Ball grid array

4.2.573.1 body

The body is an inverse relationship that specifies the Package_body (see 4.2.574) that specifies the Package as its associated_definition. There shall be no more than one Package_body for a Package.

4.2.573.2 cutout_required

The cutout_required specifies a derived value of TRUE or FALSE, depending on the existence of the optional attribute maximum_body_height_below_seating_plane. If the attribute exists, the value is TRUE. If the attribute is not present, the value is FALSE.

4.2.573.3 least_lead_length_below_seating_plane

The least_lead_length_below_seating_plane is a Length_data_element (see 4.2.511) that specifies the smallest lead length below the seating plane. The least_lead_length_below_seating_plane need not be specified for a particular Package.

4.2.573.4 least_material_condition_centroid_location

The least_material_condition_centroid_location specifies a Cartesian_point that is the location of the center of mass for the minimum material state for the Package. The least_material_condition_centroid_location need not be specified for a Package.

4.2.573.5 maximum_body_clearance_above_seating_plane

The maximum_body_clearance_above_seating_plane specifies a Length_data_element that is the maximum length between the body and the top of the seating_plane. A maximum_body_clearance_above_seating_plane need not be specified for a Package.

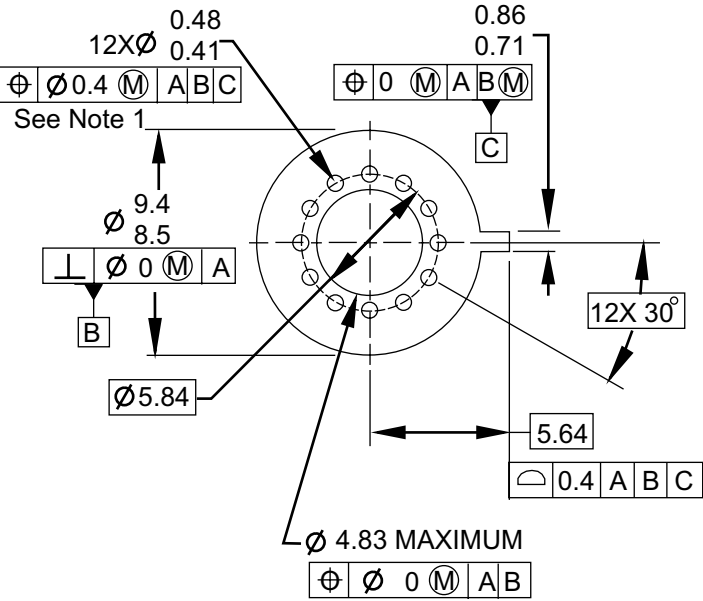
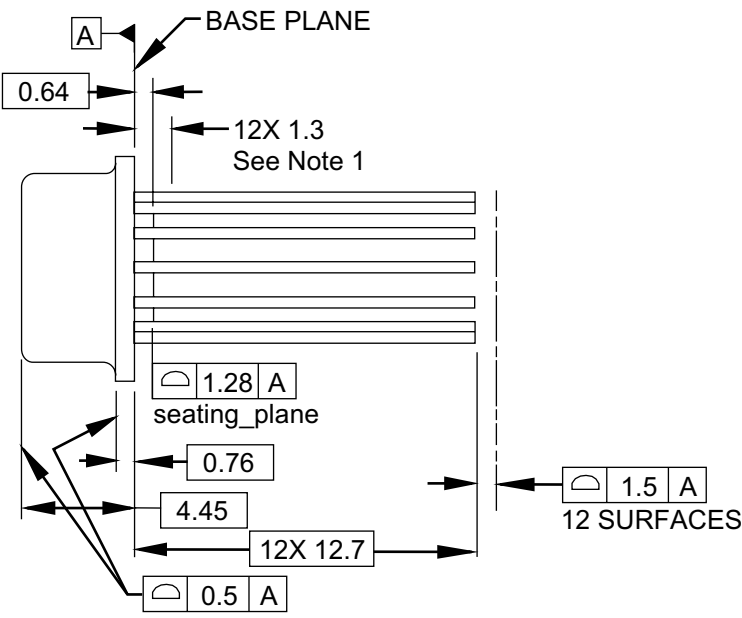
4.2.573.6 maximum_body_clearance_below_seating_plane

The maximum_body_clearance_below_seating_plane specifies a Length_data_element that is the maximum length between the body and the bottom of the seating_plane. A maximum_body_clearance_below_seating_plane need not be specified for a Package.

4.2.573.7 maximum_body_height_above_seating_plane

The maximum_body_height_above_seating_plane is a Length_data_element (see 4.2.511) that specifies the maximum height of the Package above the seating plane. Since there is no attribute describing the maximum terminal height above the seating plane, this attribute shall be interpreted as the furthest point away from the seating plane of either the body or of any Package_terminal (see 4.2.581) above the seating plane.

EXAMPLE 1 In the case of package to be mounted in a cutout there may only be the surface mounted leads protruding above the seating plane.



Note 1 : Positional tolerance of leads applies only within distance above base plane specified.

Figure 69 – T099

4.2.573.8 maximum_body_height_below_seating_plane

The `maximum_body_height_below_seating_plane` is a `Length_data_element` (see 4.2.511) that specifies the maximum height of the Package below the seating plane. The `maximum_body_height_below_seating_plane` need not be specified for a particular Package.

4.2.573.9 maximum_lead_length_below_seating_plane

The `maximum_lead_length_below_seating_plane` is a `Length_data_element` (see 4.2.511) that specifies the maximum lead length below the seating plane. The `maximum_lead_length_below_seating_plane` need not be specified for a particular Package.

4.2.573.10 maximum_material_condition_centroid_location

The `maximum_material_condition_centroid_location` specifies a `Cartesian_point` that is the location of the center of mass for the maximum material state for the Package. The `maximum_material_condition_centroid_location` need not be specified for a Package.

4.2.573.11 minimum_body_clearance_above_seating_plane

The `minimum_body_clearance_above_seating_plane` specifies a `Length_data_element` (see 4.2.511) that is the minimum length between the body and the seating plane. A `minimum_body_clearance_below_seating_plane` need not be specified for a Package.

4.2.573.12 minimum_body_clearance_below_seating_plane

The `minimum_body_clearance_below_seating_plane` specifies a `Length_data_element` that is the minimum length between the body and the seating plane. A `minimum_body_clearance_below_seating_plane` need not be specified for a Package.

4.2.573.13 package_accesses

The `package_accesses` is an inverse relationship that specifies that the existence of the Package is dependent on the existence of the `Package_terminal` (see 4.2.581) that specifies the Package as its `associated_definition`. There shall be zero or more `Package_terminal` for a Package.

4.2.573.14 package_polarity_indication

The `package_polarity_indication` is an inverse relationship that specifies the `Part_feature` (see 4.2.609) that specifies the Package as its `associated_definition`. There shall be no more than one `Part_feature` for a Package in this role.

4.2.573.15 package_primary_orientation_feature

The package_primary_orientation_feature is an inverse relationship that specifies the Part_feature (see 4.2.609) that specifies the Package as its associated_definition. There shall be no more than one Part_feature for a Package in this role.

4.2.573.16 interface_plane

The interface_plane is an inverse relationship that specifies the Seating_plane (see 4.2.802) that specifies the Package as its scope. The seating_plane simulates the interface surface of an external device.

NOTE The interface_plane is typically used for connectors, but may be used for modules.

EXAMPLE 1 A use of interface_plane is in the case of a fiber-optic transceiver that is mounted on a PCB but has a right angle connector to interface with a fiber.

4.2.573.17 package_seating_plane

The package_seating_plane is an inverse relationship that specifies the Seating_plane (see 4.2.802) that specifies the Package as its scope. The seating_plane simulates the mounting surface for the terminals in the case of surface mounted parts and in the case of through-hole technology, an arbitrarily established mounting surface. Terminals are required to approach the seating_plane from the top, which is the side where the values on the Z axis are positive. Package installation is defined to be only from the top of the seating_plane in order to reduce orientation errors between two-dimensional and three-dimensional models in this part of ISO 10303.

4.2.573.18 package_secondary_orientation_feature

The package_secondary_orientation_feature is an inverse relationship that specifies the Part_feature (see 4.2.609) that specifies the Package as its associated_definition. There shall be no more than one Part_feature for a Package in this role.

4.2.573.19 package_tertiary_orientation_feature

The package_tertiary_orientation_feature is an inverse relationship that specifies the Part_feature (see 4.2.609) that specifies the Package as its associated_definition. There shall be no more than one Part_feature for a Package in this role.

4.2.573.20 mounting_technology

The mounting_technology specifies a value of surface_mount or through_hole for the Package. Surface_mount indicates that the Package_terminals have a Connection_zone that is congruent with the package_seating_plane and through_hole indicates the terminals penetrate the package_seating_plane.

Constraints:

WR1: Only a Primary_orientation_feature shall be in the role of package_primary_orientation_feature.

WR2: Only a Secondary_orientation_feature shall be in the role of package_secondary_orientation_feature.

WR3: Only a Polarity_indication_feature shall be in the role of package_polarity_indication.

WR4: The package_primary_orientation_feature shall not be the package_secondary_orientation_feature.

WR5: The package_primary_orientation_feature shall not be the package_tertiary_orientation_feature.

WR6: The package_tertiary_orientation_feature shall not be the package_secondary_orientation_feature.

WR7: There shall be a package category associated with the product the package is a definition for.

NOTE EXPRESS specification:

*)

ENTITY package

SUPERTYPE OF (altered_package)

SUBTYPE OF (physical_unit_usage_view);

mounting_technology	: mounting_technology_type;
maximum_seating_plane_installation_offset	: OPTIONAL length_data_element;
surface_mount_lead_form	: OPTIONAL lead_form_type;
nominal_mounting_lead_pitch	: OPTIONAL length_data_element;
nominal_mounting_lead_span	: OPTIONAL length_data_element;
least_material_condition_centroid_location	: OPTIONAL cartesian_point;
maximum_material_condition_centroid_location	: OPTIONAL cartesian_point;
maximum_body_height_above_seating_plane	: length_data_element;
maximum_body_height_below_seating_plane	: OPTIONAL length_data_element;
maximum_body_clearance_above_seating_plane	: OPTIONAL length_data_element;
maximum_body_clearance_below_seating_plane	: OPTIONAL length_data_element;
minimum_body_clearance_above_seating_plane	: OPTIONAL length_data_element;
minimum_body_clearance_below_seating_plane	: OPTIONAL length_data_element;
maximum_lead_length_below_seating_plane	: OPTIONAL length_data_element;
least_lead_length_below_seating_plane	: OPTIONAL length_data_element;
of_datum_reference_frame	: OPTIONAL datum_reference_frame;
of_defining_datum_system_relationship	: OPTIONAL SET [1:3] OF
	datum_feature_usage_in_datum_system_relationship;
of_derived_datum_system_defining_relationship	: OPTIONAL SET [1:3] OF
	datum_usage_in_datum_system_relationship;

DERIVE

(*

The total vertical extent of the package calculation:

*)

```

maximum_package_total_vertical_extent : length_data_element :=
  max_data_element(
    add_data_element(maximum_body_height_above_seating_plane,
      maximum_body_height_below_seating_plane),
    add_data_element(maximum_body_height_above_seating_plane,
      maximum_lead_length_below_seating_plane));

```

(*

A cutout is required for the package if the package body extends below the seating plane. This can be determined by the existence of a value for the maximum_body_height_below_seating_plane.

*)

```

cutout_required : BOOLEAN :=
  (maximum_body_height_below_seating_plane\data_element.measure_value &#62; 0.0);
maximum_installed_height : length_data_element :=
  add_data_element(maximum_body_height_above_seating_plane,
    maximum_seating_plane_installation_offset);

```

INVERSE

```

body : SET [0:1] OF package_body FOR associated_definition;
package_seating_plane :

```

```

        non_feature_shape_element FOR scope;
interface_plane
    :
    SET[0:1] OF non_feature_shape_element FOR scope;
package_accesses : SET [0:?] OF package_terminal FOR
    associated_definition;
package_polarity_indication : SET [0:1] OF part_feature FOR
    associated_definition;
package_primary_orientation_feature          : SET [0:1] OF part_feature FOR
associated_definition;
package_secondary_orientation_feature        : SET [0:1] OF part_feature FOR
associated_definition;
package_tertiary_orientation_feature         : SET [0:1] OF part_feature FOR
associated_definition;
WHERE
    WR1: NOT(EXISTS(package_primary_orientation_feature)) OR
    ('AP210_ARM.PRIMARY_ORIENTATION_FEATURE' IN TYPEOF(package_primary_orientation_feature));
    WR2: NOT(EXISTS(package_secondary_orientation_feature)) OR
    ('AP210_ARM.SECONDARY_ORIENTATION_FEATURE' IN TYPEOF(package_secondary_orientation_feature));
    WR3: NOT(EXISTS(package_tertiary_orientation_feature)) OR
    ('AP210_ARM.TERTIARY_ORIENTATION_FEATURE' IN TYPEOF(package_tertiary_orientation_feature));
    (* orientation features shall be referenced by the
    datum_feature_usage_in_datum_system_relationship
    and shall be datum_features that establish the datum_reference_frame by establishing from
    one to 3 datum_system.
    also, the required datum_usage_in_datum_system_relationship shall be derived from
    the datum_feature_usage_in_datum_system_relationship.
    *)
    WR4: 'AP210_ARM.SEATING_PLANE' IN TYPEOF(package_seating_plane);
    WR5: 'AP210_ARM.SEATING_PLANE' IN TYPEOF(interface_plane);
    WR6: NOT(mounting_technology = surface_mount) OR
        (NOT(EXISTS(maximum_lead_length_below_seating_plane) OR
            EXISTS(least_lead_length_below_seating_plane)));
    WR7: NOT(EXISTS(maximum_lead_length_below_seating_plane)) OR
        (maximum_lead_length_below_seating_plane\data_element.measure_value &#62; 0.0);
    WR8: NOT(EXISTS(least_lead_length_below_seating_plane)) OR
        (least_lead_length_below_seating_plane\data_element.measure_value &#62; 0.0);
    WR9: (NOT(mounting_technology = surface_mount) AND
        NOT(EXISTS(surface_mount_lead_form))) OR
        ((mounting_technology = surface_mount) AND
        (EXISTS(surface_mount_lead_form)));
    WR10: package_seating_plane\shape_element.name = 'package seating plane';
    WR11: NOT(EXISTS(interface_plane[1])) OR
        (interface_plane\shape_element.name = 'interface plane');
    WR12: NOT(EXISTS(interface_plane[1])) OR
        (package_seating_plane &#60;&#62; interface_plane[1]);
END_ENTITY; -- package
(*

```

4.2.574 Package_body

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_body is a type of Part_feature (see 4.2.609) that is the encasement used in the realization of a Package (see 4.2.573). The observer view shall be above the seating plane and shall be higher than the highest point of the Package_body.

The data associated with a Package_body are the following:

- associated_definition;
- body_bottom_surface;
- body_material;
- body_top_surface.

4.2.574.1 associated_definition

The associated_definition specifies the role of the Package (see 4.2.573) for the Package_body.

4.2.574.2 body_bottom_surface

The body_bottom_surface specifies the role of a Package_body_bottom_surface (see 4.2.575) for the Package_body. A body_bottom_surface need not be specified for a Package_body.

4.2.574.3 body_material

The body_material specifies the Ee_material (see 4.2.318) used in the manufacture of the Package_body.

4.2.574.4 body_top_surface

The body_top_surface specifies the role of a Package_body_top_surface (see 4.2.579) for the Package_body. A body_top_surface need not be specified for a Package_body.

NOTE EXPRESS specification:

```

*)
ENTITY package_body
  SUBTYPE OF (part_feature);
    body_material : ee_material;
    SELF\part_feature.associated_definition : package;
WHERE
  WR1: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') |
    ('AP210_ARM.' +
    'PACKAGE_BODY_TOP_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR2: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') |
    ('AP210_ARM.' +
    'PACKAGE_BODY_BOTTOM_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
  WR3: SIZEOF(QUERY ( pf &#60;* USEDIN (SELF,
    'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') |
    ('AP210_ARM.' +
    'PACKAGE_BODY_EDGE_SURFACE' IN TYPEOF (pf))))
    &#60;= 1;
END_ENTITY; -- package_body

--tt tc1wd9 added
(

```

4.2.575 Package_body_bottom_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A package_body_bottom_surface is a type of package_body_surface (see 4.2.578) that is not visible from an observer located so as to place the component on a substrate.

NOTE EXPRESS specification:

```

*)
ENTITY package_body_bottom_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_bottom_surface

--tt tc1wd9 modified
(

```

4.2.576 Package_body_edge_segment_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_body_edge_segment_surface is a type of Package_body_surface (see 4.2.578).

NOTE EXPRESS specification:

```

*)
ENTITY package_body_edge_segment_surface
  SUBTYPE OF (package_body_surface);
    composed_surface : package_body_edge_surface;
    start_vertex      : edge_segment_vertex;
    end_vertex        : edge_segment_vertex;
WHERE
  WR1: start_vertex :&#60;&#62;; end_vertex;
END_ENTITY; -- package_body_edge_segment_surface

--tt tc1wd9 added
( *
```

4.2.577 Package_body_edge_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_body_edge_surface is a type of Package_body_surface (see 4.2.578).

NOTE EXPRESS specification:

```

*)
ENTITY package_body_edge_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_edge_surface

( *
```

4.2.578 Package_body_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_body_surface is either a Package_body_top_surface (see 4.2.579) or a Package_body_bottom_surface (see 4.2.575). A Package_body_surface relationship to the seating plane may be explicitly established, helping to unambiguously orient the Packaged_part (see 4.2.591).

The data associated with a Package_body_surface are the following:

— associated_package_body.

4.2.578.1 associated_package_body

The associated_package_body specifies the role of the Package_body (see 4.2.574) for the Package_body_surface.

NOTE EXPRESS specification:

```
*)
ENTITY package_body_surface
  ABSTRACT SUPERTYPE OF (ONEOF (
    package_body_bottom_surface,
    package_body_top_surface,
    package_body_edge_segment_surface,
    package_body_edge_surface))
  SUBTYPE OF (part_feature);
  associated_package_body : package_body;
END_ENTITY; -- package_body_surface
```

(*

4.2.579 Package_body_top_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A package_body_top_surface is a type of Package_body_surface (see 4.2.578) that is visible from any observer located so as to place the component on a substrate.

NOTE EXPRESS specification:

```
*)
ENTITY package_body_top_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_top_surface
```

(*

4.2.580 Package_external_reference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_external_reference is a type of Managed_design_object (see 4.2.524). A Package_external_reference is an unambiguous reference to the definition of a Package (see 4.2.573) in an external system.

NOTE Package_external_reference is included to permit references to Packaged_part_terminal (see 4.2.594) without requiring that a geometric representation be present for the terminal.

The data associated with a Package_external_reference are the following:

- design_owner;
- part_number;
- revision_code;
- product_definition_id.

4.2.580.1 design_owner

The design_owner specifies one role of a string for the Package_external_reference. The design_owner controls the assignment of part_number for the Package being referenced.

4.2.580.2 part_number

The part_number specifies one role of a string for the Package_external_reference. The part_number represents an identifier unique within the organization for packages.

4.2.580.3 revision_code

The revision_code specifies one role of a string for the Package_external_reference. The revision_code represents a version of the package, and together with part_number is unique within the organization for packages.

4.2.580.4 product_definition_id

The product_definition_id specifies one role of a string for the Package_external_reference. The product_definition_id represents the identification of the definition of the Package being referenced.

NOTE EXPRESS specification:

```

*)
ENTITY package_external_reference
  SUBTYPE OF (managed_design_object);
    design_owner      : STRING;
    part_number       : STRING;
    revision_code     : STRING;
    product_definition_id : STRING;
END_ENTITY; -- package_external_reference

```

(*

4.2.581 Package_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Package_terminal is a type of Part_feature (see 4.2.609). A Package_terminal may be one or more of a Guided_wave_terminal (see 4.2.427), a Altered_package_terminal (see 4.2.9), a Wire_terminal (see 4.2.963), or a Primary_reference_terminal (see 4.2.718). When the case or package body is used as a terminal then Package_terminal shall be populated to specify the feature of the Package_body (see 4.2.574) that serves that purpose. The Package_terminal shall be in a sequence, with the initial member being Primary_reference_terminal, and each member that is not a Primary_reference_terminal shall reference a previous member in the sequence using the precedent_feature attribute inherited from Part_feature. The seating_plane_zone shall not exist if the seating_plane_intersection attribute is 'does not intersect'. The seating_plane_intersection attribute shall not be 'does not intersect' if the seating_plane_zone attribute exists.

NOTE The Seating_plane referenced indirectly by this terminal may be in the role of interface_plane for the Package referenced by associaed_definition.

EXAMPLE 1 Figure 70 illustrates a cutaway view of a terminal and shows the characteristics related to a terminal.

The data associated with a Package_terminal are the following:

- associated_definition;
- external_connection_zone;
- feature_shape;
- internal_connection_zone;
- seating_plane_intersection;
- maximum_terminal_diametrical_extent;
- minimum_terminal_diametrical_extent;
- model_representation;
- terminal_characteristic;
- terminal_core_material;
- terminal_surface_material.

4.2.581.1 associated_definition

The associated_definition specifies the role of the Package (see 4.2.573) for the Package_terminal.

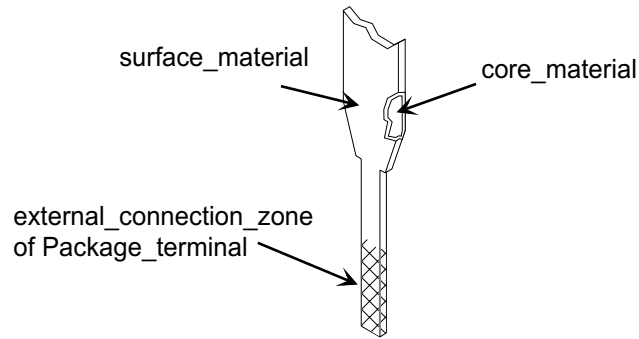


Figure 70 – Package_terminal

4.2.581.2 external_connection_zone

The `external_connection_zone` specifies the `Connection_zone` (see 4.2.185) on the outside surface of the `Package_terminal`. There may be more than one `Connection_zone` for a `Package_terminal`. The `external_connection_zone` is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the `external_connection_zone` shall be supplied.

NOTE The `Feature_shape_occurrence` (see ??) specified by the terminal or mounting feature may be the same `Feature_shape_occurrence` that is specified by the associated `Connection_zone` (see 4.2.185). In this case the whole terminal is the `Connection_zone`.

4.2.581.3 feature_shape

The `feature_shape` specifies the role of the `Feature_shape_occurrence` (see ??) for the `Package_terminal`. There shall be one or more `Feature_shape_occurrence` for the `Package_terminal`.

4.2.581.4 internal_connection_zone

The `internal_connection_zone` specifies the `Connection_zone` (see 4.2.185) on the inside portion of the `Package_terminal`. There may be more than one `Connection_zone` in the role of `internal_connection_zone`. The `internal_connection_zone` need not be specified for a particular `Package_terminal`.

NOTE The `internal_connection_zone` is separate from the `external_connection_zone`.

4.2.581.5 seating_plane_intersection

The `seating_plane_intersection` specifies whether or not the shape of the `Package_terminal` intersects the shape of the seating plane to within the tolerance of the geometric context. The values allowed for the attribute are 'surface_intersection', 'through_intersection', 'does_not_intersect'.

NOTE Not all terminals need to intersect the seating plane.

4.2.581.6 maximum_terminal_diametrical_extent

The `maximum_terminal_diametrical_extent` specifies the `Length_data_element` (see 4.2.511) that represents the maximum allowed diameter of a `Package_terminal`. The `Length_data_element` need not be specified for a `Package_terminal`. The `Length_data_element` is the diameter of a circle that circumscribes the cross-sectional shape of the terminal.

4.2.581.7 minimum_terminal_diametrical_extent

The `minimum_terminal_diametrical_extent` specifies the `Length_data_element` (see 4.2.511) that represents the minimum allowed diameter of a `Package_terminal`. The `Length_data_element` need not be specified for a `Package_terminal`. The `Length_data_element` is the diameter of a circle that circumscribes the cross-sectional shape of the terminal.

4.2.581.8 model_representation

The `model_representation` specifies the `Analytical_representation` (see 4.2.22) that embodies the analysis characteristics identified as `terminal_characteristics`. The `model_representation` need not be specified for a particular `Package_terminal`. There may be more than one `Analytical_representation` for a `Package_terminal`.

4.2.581.9 terminal_characteristic

The `terminal_characteristic` specifies the `Characteristic` (see 4.2.98) of the `Package_terminal`. The `terminal_characteristic` need not be specified for a particular `Package_terminal`. There may be more than one `Characteristic` for a `Package_terminal`.

4.2.581.10 terminal_core_material

The `terminal_core_material` specifies the role of the `Material_definition` (see 4.2.531) for the `Package_terminal` core material. The `terminal_core_material` is optional and need not be supplied.

4.2.581.11 terminal_surface_material

The `terminal_surface_material` specifies the role of the `Material_definition` (see 4.2.531) for the outer surface of the `Package_terminal` surface material.

Constraints:

WR1: The `Package` instance specified by `associated_definition` shall have a cartesian coordinate system frame of reference.

NOTE EXPRESS specification:

```

*)
ENTITY package_terminal
  SUPERTYPE OF (((altered_package_terminal ANDOR guided_wave_terminal)
    ANDOR wire_terminal) ANDOR primary_reference_terminal)
  SUBTYPE OF (part_feature);
  SELF\part_feature.associated_definition : package;
  external_connection_zone                : OPTIONAL SET [1:?] OF connection_zone;
  internal_connection_zone                : OPTIONAL SET [1:?] OF connection_zone;
  seating_plane_intersection              : seating_plane_intersection_type;
  terminal_surface_material               : material_definition;
  model_representation                   : OPTIONAL SET [1:?] OF
    analytical_representation;
  terminal_characteristic                 : OPTIONAL SET [1:?] OF characteristic;
  terminal_core_material                 : OPTIONAL material_definition;
  minimum_terminal_diametrical_extent    : OPTIONAL length_data_element;
  maximum_terminal_diametrical_extent    : OPTIONAL length_data_element;
  seating_plane_zone                     :
    OPTIONAL connection_zone_interface_plane_relationship;
  DERIVE
    mates_with_substrate : BOOLEAN := (EXISTS(seating_plane_zone)) AND
      (associated_definition.package_seating_plane :=:
        seating_plane_zone.interface_plane);
  INVERSE
    SELF\part_feature.feature_shape      : SET [1:?] OF usage_concept_usage_relationship;
    associated_usage;
  WHERE
    WR1: NOT(EXISTS(seating_plane_zone)) OR
      (SIZEOF(QUERY( cz &#60;* external_connection_zone |
        (seating_plane_zone.derived_zone = cz)
      )) = 1);
    WR2: NOT(EXISTS(seating_plane_zone)) OR
      (SIZEOF(QUERY( cz &#60;* external_connection_zone |
        (seating_plane_zone.mating_zone = cz)
      )) = 1);
    WR3: NOT(EXISTS(seating_plane_zone)) OR
      ((associated_definition.package_seating_plane :=:
        seating_plane_zone.interface_plane) OR
        (associated_definition.interface_plane[1] :=:
        seating_plane_zone.interface_plane));
    WR4: NOT(EXISTS(seating_plane_zone)) OR
      ((seating_plane_intersection = surface_intersection) OR
        (seating_plane_intersection = through_intersection));
    WR5: NOT((seating_plane_intersection = does_not_intersect) AND
      (EXISTS(seating_plane_zone)));
  END_ENTITY; -- package_terminal

( *
```

4.2.582 Package_terminal_external_reference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Package_terminal_external_reference` is a type of `Managed_design_object` (see 4.2.524). A `Package_terminal_external_reference` is an unambiguous reference to a `Package_terminal` (see 4.2.581) in an external system.

EXAMPLE 1 Items in a `Physical_connectivity_definition` (see 4.2.652) may need to reference terminals in a library without duplicating the geometric context.

The data associated with a `Package_terminal_external_reference` are the following:

- `associated_definition`;
- `name`.

4.2.582.1 associated_definition

The `associated_definition` specifies the role for a `Package_external_reference` (see 4.2.580) for the `Package_terminal_external_reference`.

4.2.582.2 name

The `name` specifies the role of a string for the `Package_terminal_external_reference`. The `name` shall be interpreted as the `name` attribute inherited by `Package_terminal` from `Shape_element`.

Constraints:

UR1: Each member of `Package_terminal_external_reference` shall have a unique combination of attributes.

NOTE EXPRESS specification:

```
*)
ENTITY package_terminal_external_reference
  SUBTYPE OF (managed_design_object);
    associated_definition : package_external_reference;
    name                  : STRING;
  UNIQUE
    UR1 : associated_definition, name;
END_ENTITY; -- package_terminal_external_reference

( *
```

4.2.583 Package_terminal_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Package_terminal_occurrence` is a type of `Managed_design_object` (see 4.2.524). A `Package_terminal_occurrence` is an occurrence of a `Package_terminal` (see 4.2.581) in a `Packaged_component` (see 4.2.585) for the case where the equivalent `Packaged_part_terminal` (see 4.2.594) is not populated. This application object is provided to specify the cases where no `Packaged_part_terminal` is defined for a `Package_terminal` because there is no access to functionality specified through that `Package_terminal`. There is a requirement to support the fact that the `Assembly_joint` (see 4.2.41) connects the realized terminal of the component to the realized terminal of an `Interconnect_module_component` (see 4.2.442) or to other members of `Assembly_component`.

The data associated with a `Package_terminal_occurrence` are the following:

- `associated_component`;
- `definition`.

4.2.583.1 `associated_component`

The `associated_component` specifies the role of the `Packaged_component` (see 4.2.585) for the `Package_terminal_occurrence`.

4.2.583.2 `definition`

The `definition` specifies the role of the `Package_terminal` (see 4.2.581) that has no defined electrical functionality for the `Package_terminal_occurrence`.

NOTE EXPRESS specification:

```

*)
ENTITY package_terminal_occurrence
  SUBTYPE OF (managed_design_object);
    definition          : package_terminal;
    associated_component : packaged_component;
END_ENTITY; -- package_terminal_occurrence

--tt tc1 wd9 add
( *
```

4.2.584 `Package_terminal_surface_constituent_relationship`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Package_terminal_surface_constituent_relationship` is a type of `Shape_element_constituent_relationship` (see 4.2.813).

NOTE EXPRESS specification:

```

*)
ENTITY package_terminal_surface_constituent_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relatng_shape_element : package_body_surface;
  SELF\shape_element_relationship.related_shape_element : package_terminal;
END_ENTITY; -- package_terminal_surface_constituent_relationship

( *
```

4.2.585 Packaged_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Packaged_component is a type of Assembly_component (see 4.2.34). A Packaged_component may be a Packaged_connector_component (see 4.2.588) or additionally a Routed_packaged_component (see ??). A Packaged_component is an instantiation of a Packaged_part (see 4.2.591) at a given location within the context of a printed circuit assembly.

The data associated with a Packaged_component are the following:

- access_mechanism;
- selected_package_alterate;
- usage_definition.

4.2.585.1 access_mechanism

The access_mechanism is an inverse relationship that specifies that the existence of the Packaged_component is dependent on the existence of the Packaged_component_join_terminal (see 4.2.586) that specifies the Packaged_component as its associated_component. There shall be one or more Packaged_component_join_terminal for a Packaged_component.

4.2.585.2 selected_package_alterate

The selected_package_alterate specifies the role of the Altered_package (see 4.2.8) for the Packaged_component. The selected_package_alterate need not be specified for a particular Packaged_component.

4.2.585.3 usage_definition

The usage_definition specifies the role of the Packaged_part (see 4.2.591) for the Packaged_component.

Constraints:

WR1: If `selected_package_alternate` exists, then the `selected_package_alternate` shall be in the role of `used_package` for the `Packaged_part` in the role of `usage_definition` for the `Packaged_component`.

NOTE EXPRESS specification:

```

*)
ENTITY packaged_component
  SUPERTYPE OF (packaged_connector_component)
  --tt tc1 wd11 modify
  SUBTYPE OF (physical_component);
    SELF\assembly_component.usage_definition : packaged_part;
    selected_package_alternate                : OPTIONAL altered_package;
  INVERSE
    access_mechanism : SET [1:?] OF packaged_component_join_terminal FOR
                          associated_component;
END_ENTITY; -- packaged_component

( *
```

4.2.586 Packaged_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_component_join_terminal` is a type of `Physical_component_terminal` (see 4.2.650). A `Packaged_component_join_terminal` may be a `Movable_packaged_component_join_terminal` (see 4.2.572). A `Packaged_component_join_terminal` is one termination for a `Packaged_component` (see 4.2.585). The terminal provides access to the electrical functionality of the packaged part being terminated.

EXAMPLE 1 Figure 71 illustrates a detail of a assembly drawing that shows a microcircuit plugged into a socket, and the role of the `Packaged_component_join_terminal`.

The data associated with a `Packaged_component_join_terminal` are the following:

- `associated_component`;
- `definition`.

4.2.586.1 associated_component

The `associated_component` specifies the `Packaged_component` (see 4.2.585) being accessed by the `Packaged_component_join_terminal`.

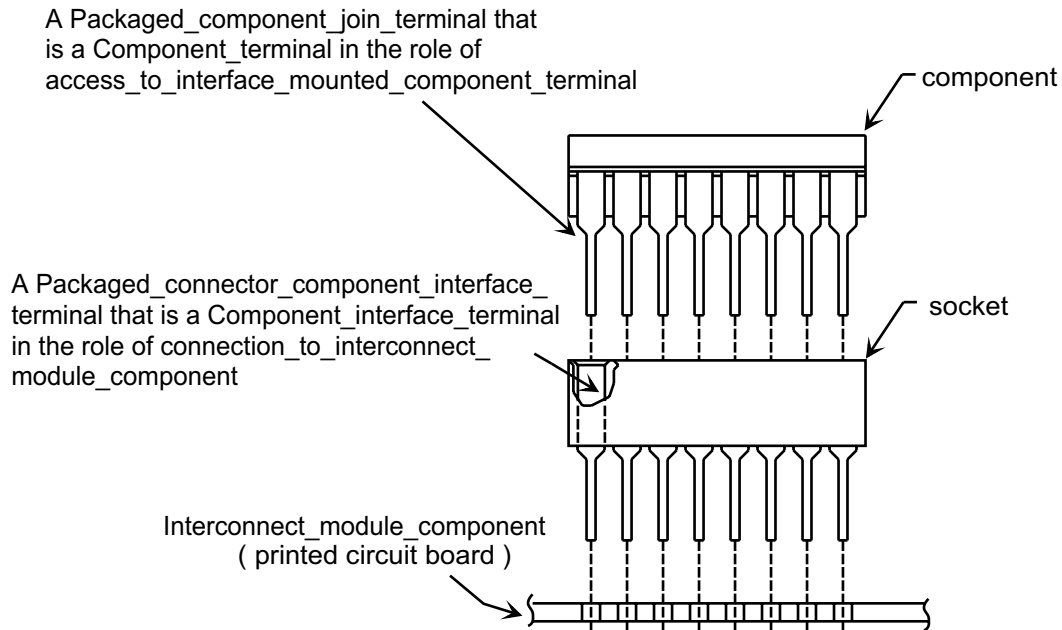


Figure 71 – Packaged_component_join_terminal

4.2.586.2 definition

The definition specifies the role of the Packaged_part_join_terminal (see 4.2.593) for the Packaged_component_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY packaged_component_join_terminal
  SUPERTYPE OF (movable_packaged_component_join_terminal)
  SUBTYPE OF (physical_component_terminal);
  SELF\component_feature.definition          : packaged_part_join_terminal;
  SELF\component_feature.associated_component : packaged_component;
END_ENTITY; -- packaged_component_join_terminal

```

(*

4.2.587 Packaged_connector

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_connector` is a type of `Packaged_part` (see 4.2.591) whose functionality between certain terminals is defined to be a short by this part of ISO 10303. The short shall be between the interface terminal and the join terminal specified by a member of `Packaged_connector_terminal_relationship` (see 4.2.590). The short is a logical abstraction for CAD purposes. The detailed functional properties of the connector are defined in the connector specification.

NOTE One use for a `Packaged_connector` is to provide access to the physical network of the printed circuit assembly from an external source.

EXAMPLE 1 Figure 72 illustrates a `Packaged_connector` and its usage in a design context as a `Packaged_connector_component` (see 4.2.588).

The data associated with a `Packaged_connector` are the following:

— `characterized_connector_terminal_relationship`.

4.2.587.1 `characterized_connector_terminal_relationship`

The `characterized_connector_terminal_relationship` is an inverse relationship that specifies that the existence of the `Packaged_connector` is dependent on the existence of the `Packaged_connector_terminal_relationship` (see 4.2.590) that specifies the `Packaged_connector` as its connector. There shall be one or more `Packaged_connector_terminal_relationship` for a `Packaged_connector`.

NOTE EXPRESS specification:

```
*)
ENTITY packaged_connector
  SUBTYPE OF (packaged_part);
  INVERSE
    characterized_connector_terminal_relationship :
      SET [1:?] OF packaged_connector_terminal_relationship FOR connector;
END_ENTITY; -- packaged_connector
(*
```

4.2.588 `Packaged_connector_component`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_connector_component` is a type of `Packaged_component` (see 4.2.585) that is an occurrence of a `Packaged_connector` (see 4.2.587).

EXAMPLE 1 Figure 73 illustrates a member of `Packaged_connector_component` that is a socket.

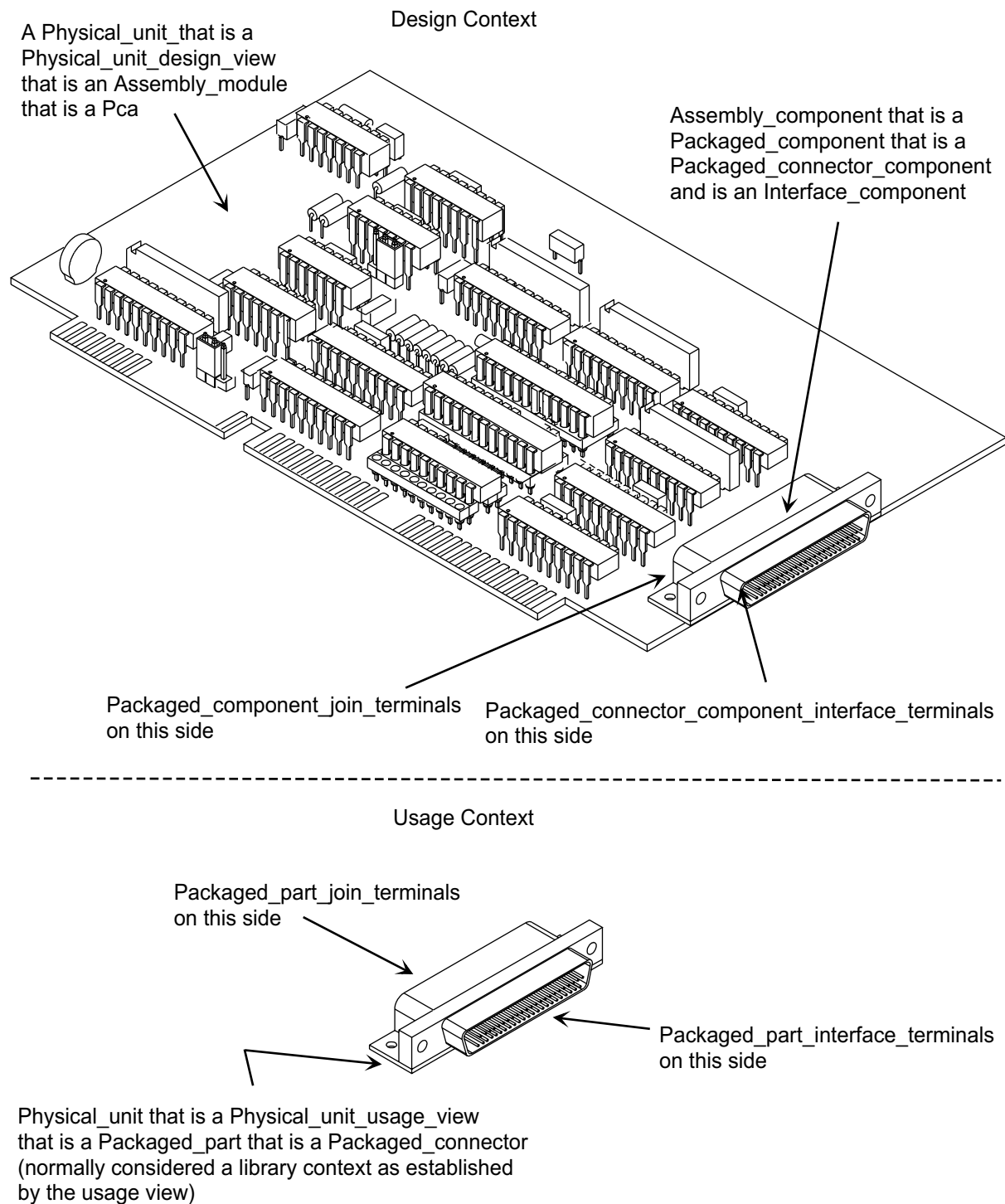


Figure 72 – Packaged_connector

The data associated with a `Packaged_connector_component` are the following:

- `interface_access_mechanism`;
- `usage_definition`.

4.2.588.1 `interface_access_mechanism`

The `interface_access_mechanism` is an inverse relationship that specifies that the existence of the `Packaged_connector_component` is dependent on the existence of the `Packaged_connector_component_interface_terminal` (see 4.2.589) that specifies the `Packaged_connector_component` as its `associated_component`. There shall be one or more `Packaged_connector_component_interface_terminal` for a `Packaged_connector_component`.

4.2.588.2 `usage_definition`

The `usage_definition` specifies the role of the `Packaged_connector` (see 4.2.587) for the `Packaged_connector_component`.

NOTE EXPRESS specification:

```

*)
ENTITY packaged_connector_component
--tt tc1 wd9 modified
  SUBTYPE OF (interface_component, packaged_component);
  SELF\assembly_component.usage_definition : packaged_connector;
  INVERSE
    interface_access_mechanism : SET [1:?] OF
                                     packaged_connector_component_interface_terminal FOR
                                     associated_component;
END_ENTITY; -- packaged_connector_component

( *
```

4.2.589 `Packaged_connector_component_interface_terminal`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_connector_component_interface_terminal` is a type of `Component_interface_terminal` (see 4.2.125). The `Packaged_connector_component_interface_terminal` indicates explicitly that the object is an interface terminal of a connector and that the terminal will never be connected to the circuit board or substrate.

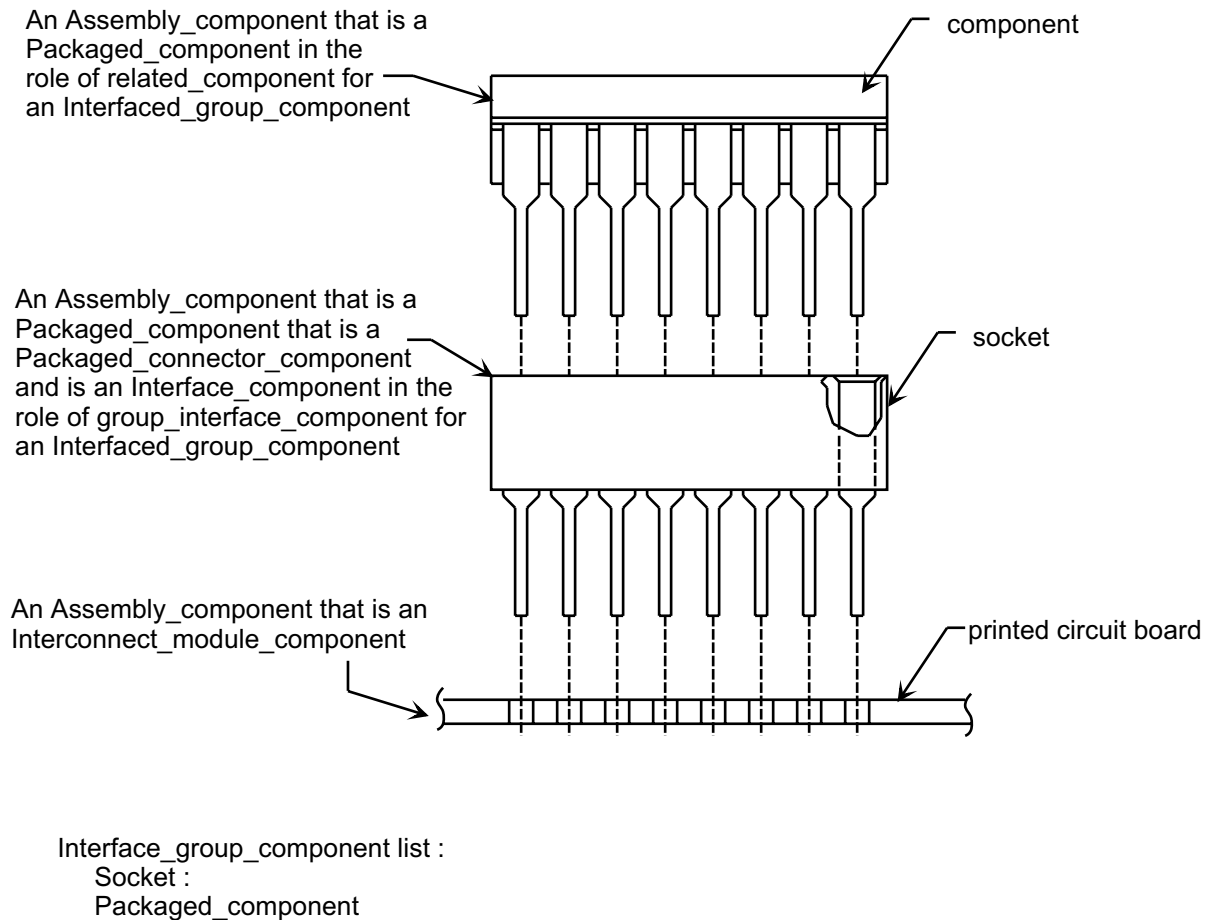


Figure 73 – Packaged_connector_component

NOTE In the context of system simulation and test, a `Packaged_connector_component_interface_terminal` may represent a cabling or external connection interface. Further, it provides the exact geometric location of the terminal, whereby the exact geometric location of a mating terminal may be checked to determine whether the physical connection zones of the two interface terminals mate properly.

The data associated with a `Packaged_connector_component_interface_terminal` are the following:

- `associated_component`;
- `definition`.

4.2.589.1 associated_component

The `associated_component` specifies the role of the `Packaged_connector_component` (see 4.2.588) for the `Packaged_connector_component_interface_terminal`.

4.2.589.2 definition

The definition specifies the role of the `Packaged_part_interface_terminal` (see 4.2.592) for the `Packaged_connector_component_interface_terminal`.

NOTE EXPRESS specification:

```
*)
ENTITY packaged_connector_component_interface_terminal
  SUBTYPE OF (component_interface_terminal);
    SELF\component_feature.definition      : packaged_part_interface_terminal;
    SELF\component_feature.associated_component : packaged_connector_component;
END_ENTITY; -- packaged_connector_component_interface_terminal

( *
```

4.2.590 Packaged_connector_terminal_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_connector_terminal_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Packaged_connector_terminal_relationship` is the association between one interface terminal and a corresponding join terminal of the `Packaged_connector` (see 4.2.587) that specifies that the two terminals are shorted internally to the `Packaged_connector` as part of its required functionality. This relationship is an explicit statement of the inherent functionality of a connector. Therefore no separate `Functional_unit_network_definition` (see 4.2.402) or `Analytical_model` (see 4.2.14) is required to specify the nominal functionality of a connector. A `Functional_unit_usage_view` (see 4.2.410) may be supplied to support mapping the functional design to the connector join terminals.

NOTE 1 One use for this relationship is to specify the short between the terminal that is installed to the printed circuit assembly and the terminal that provides the access to the outside environment when an occurrence of `Packaged_connector` is installed in the assembly.

NOTE 2 Installation into the next higher assembly is not necessary to achieve this relationship.

NOTE 3 There is a member of this relationship populated for every pair of terminals in the `Packaged_connector`.

The data associated with a `Packaged_connector_terminal_relationship` are the following:

- connector;
- interface_terminal;
- join_terminal.

4.2.590.1 connector

The connector specifies the Packaged_connector (see 4.2.587) that has the Package_terminal (see 4.2.581) that are defined to be shorted by the Packaged_connector_terminal_relationship.

4.2.590.2 interface_terminal

The interface_terminal specifies the role of the Packaged_part_interface_terminal (see 4.2.592) for the Packaged_connector_terminal_relationship.

4.2.590.3 join_terminal

The join_terminal specifies the role of the Packaged_part_join_terminal (see 4.2.593) for the Packaged_connector_terminal_relationship.

Constraints:

UR1: The combination of the connector and the join_terminal shall be unique within a population of Packaged_connector_terminal_relationship.

UR2: The combination of the connector and the interface_terminal shall be unique within a population of Packaged_connector_terminal_relationship.

WR1: The Packaged_part associated with the interface_terminal shall be the Packaged_connector specified by connector.

WR2: The Packaged_part associated with the join_terminal shall be the Packaged_connector specified by connector.

NOTE EXPRESS specification:

```

*)
ENTITY packaged_connector_terminal_relationship
  SUBTYPE OF (managed_design_object);
    connector          : packaged_connector;
    join_terminal      : packaged_part_join_terminal;
    interface_terminal : packaged_part_interface_terminal;
  UNIQUE
    UR1 : join_terminal, connector;
    UR2 : interface_terminal, connector;
END_ENTITY; -- packaged_connector_terminal_relationship

```

(*

4.2.591 Packaged_part

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_part` is a type of a `Physical_unit_usage_view` (see 4.2.672). A `Packaged_part` may be a `Packaged_connector` (see 4.2.587) or additionally an `Altered_packaged_part` (see 4.2.10). A `Packaged_part` describes a part that exists independently from any particular assembly on which it may be used.

NOTE 1 The `Packaged_part` Application object does not indicate that a product used as a component is in a particular product category.

NOTE 2 A `Packaged_part` sometimes is realized by installing one or more occurrences of `Bare_die` (see 4.2.59) into a `Package_body` (see 4.2.574), bonding conductors between appropriate ports on the occurrence of the `Bare_die` and `Package_terminal` (see 4.2.581) on the `Package` (see 4.2.573), filling the void between the occurrences of the `Bare_die` and the `Package_body` with an insulating potting material, and marking the realization with an identifier. See Figure 74 for an illustration of a cutaway view of a `Packaged_part`.

NOTE 3 Two `Packaged_part` models may be exchanged with this part of ISO 10303, depending on usage. One may support the exchange information that maps the functionality of the `Packaged_part` to the `bare_die`, or dies, included in the `Packaged_part`; and information that maps the terminals of those instances of `Bare_die` to the terminals of the `Packaged_part`. This mechanism provides maximum flexibility and information exchange and is characterized as a "white box" model. This model is primarily included in the standard to support design to analysis activities. Alternatively, one may map the functionality required directly to the `Packaged_part` terminals, and not populate the `Bare_die`. This is simpler, but does not provide as much information and is characterized as the "black box" model.

The data associated with a `Packaged_part` are the following:

- `access_mechanisms`;
- `implemented_function`;
- `potting_compound`;
- `used_package`.

4.2.591.1 access_mechanisms

The `access_mechanisms` is an inverse relationship that specifies that the existence of the `Packaged_part` is dependent on the existence of the `Packaged_part_terminal` (see 4.2.594) that specifies the `Packaged_part` as its associated_definition. There shall be zero or more `Packaged_part_terminal` for a `Packaged_part`.

4.2.591.2 implemented_function

The `implemented_function` specifies the role of the `Functional_unit_usage_view` (see 4.2.410) for the `Packaged_part`. The function specified shall be that mapped by the functional view terminal to physical view terminal allocation requirement. The `implemented_function` need not be specified for a particular `Packaged_part` unless the `access_mechanism` is not an empty set.

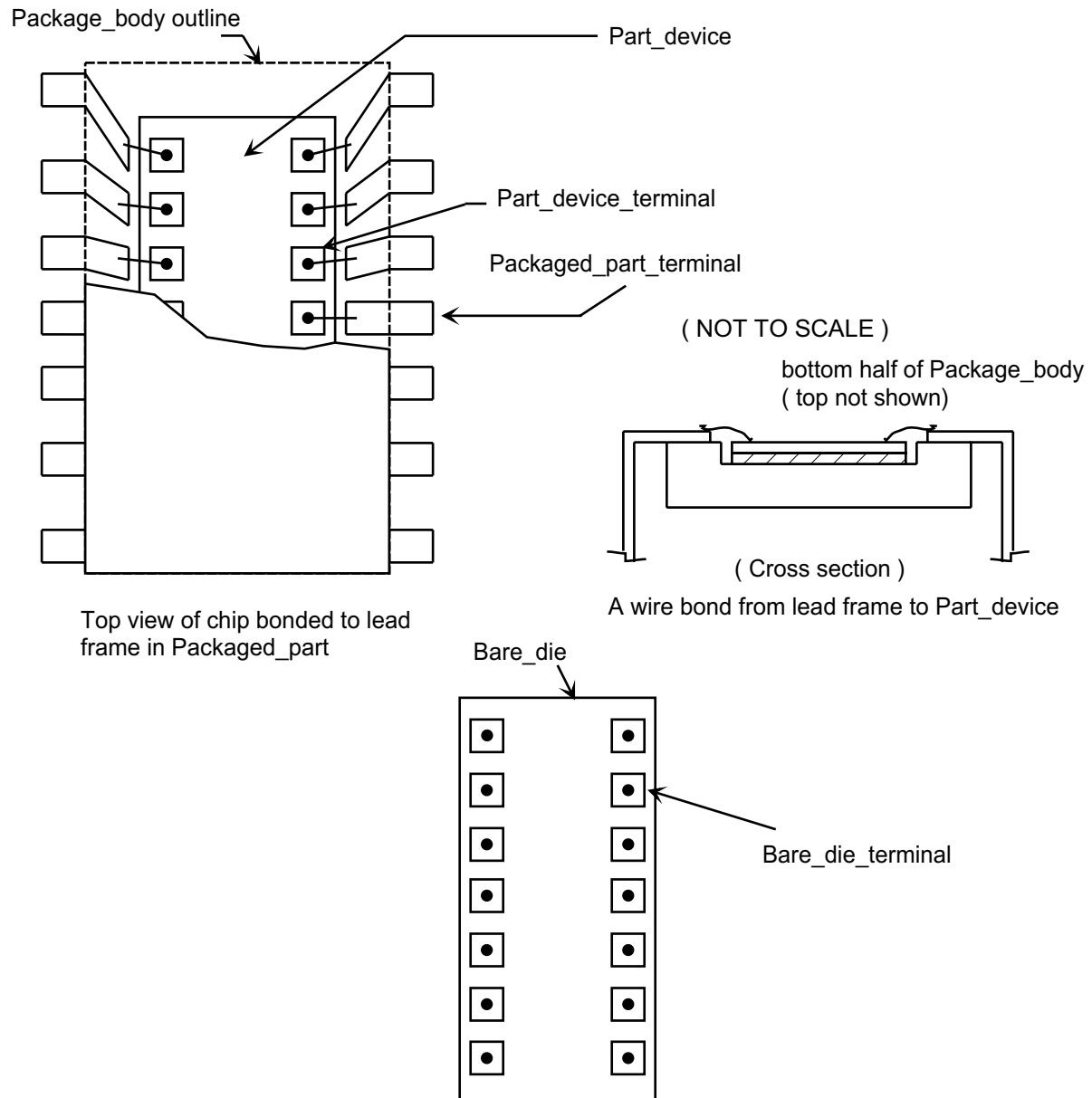


Figure 74 – Packaged_part exploded assembly detail

4.2.591.3 potting_compound

The potting_compound specifies the Ee_material (see 4.2.318) that is used to support the enclosed Bare_die (see 4.2.59) in the Packaged_part. The potting_compound need not be specified for a particular Packaged_part. The potting_compound may form the Package_body (see 4.2.574) itself for some technologies when the Package_body is molded from the potting_compound.

4.2.591.4 used_package

The used_package specifies the Package (see 4.2.573) of the Packaged_part. There may be one or more used_package for a Packaged_part. All members of used_package shall reference the same Ee_product_version.

Constraints:

WR1: The function implemented shall exist if and only if the size of the access_mechanisms is greater than zero and the size of the access_mechanisms shall be greater than zero if and only if the implemented_function exists.

WR2: The function implemented shall be the function whose terminals are mapped to the terminals of this Packaged_part.

IP1: The combination of implemented_function and used_package\ee_product_definition.version shall be unique within a population of Packaged_part. In the case where the Package_external_reference is used the combination shall be extended to include the Package_external_reference.part_number string.

NOTE EXPRESS specification:

```

*)
ENTITY packaged_part
  SUPERTYPE OF (packaged_connector ANDOR altered_packaged_part)
  SUBTYPE OF (physical_unit_usage_view);
  used_package          : SET [1:?] OF package_or_package_external_reference;
  potting_compound      : OPTIONAL ee_material;
  implemented_function  : OPTIONAL functional_unit_usage_view;
  INVERSE
    access_mechanisms : SET [0:?] OF packaged_part_terminal FOR
                        associated_definition;
WHERE
  WR1: NOT(EXISTS(implemented_function) XOR (SIZEOF(access_mechanisms) &#62; 0));
  WR2: single_package_version(SELF);
END_ENTITY; -- packaged_part
( *
```

4.2.592 Packaged_part_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_part_interface_terminal` is a type of `Packaged_part_terminal` (see 4.2.594) that is a terminal of a `Packaged_connector` (see 4.2.587). An occurrence of the `Packaged_part_interface_terminal` provides access to the functionality of the assembly into which an occurrence of a `Packaged_connector` is installed.

The data associated with a `Packaged_part_interface_terminal` are the following:

— `external_connector_function`.

4.2.592.1 `external_connector_function`

The `external_connector_function` is an inverse relationship that specifies that the existence of the `Packaged_part_interface_terminal` is dependent on the existence of the `Packaged_connector_terminal_relationship` (see 4.2.590) that specifies the `Packaged_part_interface_terminal` as its `interface_terminal`. There shall be one or more `Packaged_connector_terminal_relationship` for a `Packaged_part_interface_terminal`.

NOTE One interface terminal may be connected to several join terminals by members of `Packaged_connector_terminal_relationship`.

NOTE EXPRESS specification:

```
*)
ENTITY packaged_part_interface_terminal
  SUBTYPE OF (packaged_part_terminal);
  INVERSE
    external_connector_function : SET [1:?] OF
      packaged_connector_terminal_relationship FOR
      interface_terminal;
END_ENTITY; -- packaged_part_interface_terminal

( *
```

4.2.593 `Packaged_part_join_terminal`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_part_join_terminal` is a type of `Packaged_part_terminal` (see 4.2.594) such that an occurrence of the `Packaged_part_terminal` shall not be in the role of an external interface when the occurrence of the associated `Packaged_connector` (see 4.2.587) is a component in an assembly.

The data associated with a `Packaged_part_join_terminal` are the following:

— `internal_connector_function`.

4.2.593.1 internal_connector_function

The `internal_connector_function` is an inverse relationship that specifies the `Packaged_connector_terminal_relationship` (see 4.2.590) that specifies the `Packaged_part_join_terminal` as its `join_terminal`. There shall be no more than one `Packaged_connector_terminal_relationship` for a `Packaged_part_join_terminal`.

NOTE EXPRESS specification:

```
*)
ENTITY packaged_part_join_terminal
  SUBTYPE OF (packaged_part_terminal);
  INVERSE
    internal_connector_function : SET [0:1] OF
      packaged_connector_terminal_relationship FOR
      join_terminal;
END_ENTITY; -- packaged_part_join_terminal

(*
```

4.2.594 Packaged_part_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Packaged_part_terminal` is a type of `Part_terminal` (see 4.2.630) that describes a terminal of a `Packaged_part` (see 4.2.591). A `Packaged_part_terminal` is one of a `Packaged_part_interface_terminal` (see 4.2.592) or a `Packaged_part_join_terminal` (see 4.2.593).

The data associated with a `Packaged_part_terminal` are the following:

- `associated_definition`;
- `terminal_connection_zone`;
- `terminal_of_package`.

4.2.594.1 associated_definition

The `associated_definition` specifies the `Packaged_part` (see 4.2.591) that has the `Packaged_part_terminal`.

4.2.594.2 terminal_connection_zone

The `terminal_connection_zone` specifies the `Connection_zone` (see 4.2.185) of the `Packaged_part_terminal`. There may be more than one `Connection_zone` for a `Packaged_part_terminal`. The `terminal_connection_zone` need not be specified for a particular `Packaged_part_terminal`.

4.2.594.3 terminal_of_package

The terminal_of_package specifies the role of either the Package_terminal (see 4.2.581) or the Package_terminal_external_reference (see 4.2.582) for the Packaged_part_terminal. There may be more than one terminal_of_package. If there is more than one terminal_of_package then the member reference different Package members.

Constraints:

WR1: Each member of terminal_of_package shall specify a member of Package that is specified by the Packaged_part in the role of associated_definition as used_package. No member of Package that is specified by the Packaged_part in the role of a associated_definition as used_package shall be specified by more than one member of terminal_of_package.

WR2: The precedent_feature attribute inherited from Part_feature shall not exist.

NOTE EXPRESS specification:

```
*)
ENTITY packaged_part_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (packaged_part_interface_terminal,
    packaged_part_join_terminal))
  SUBTYPE OF (part_terminal);
    terminal_of_package :
  SET [1:?] OF package_terminal_or_package_terminal_external_reference;
  SELF\part_feature.associated_definition : packaged_part;
WHERE
  WR1: disjoint_package_terminals(SELF);
  WR2: NOT EXISTS(SELF\part_feature.precedent_feature);
END_ENTITY; -- packaged_part_terminal
(*
```

4.2.595 Padstack_definition

A Padstack_definition is a type of Ee_product_definition (see 4.2.323).

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_definition
  SUPERTYPE OF(ONEOF(
    design_layer_type_specific_padstack_definition,
    stratum_type_independent_padstack_definition,
    stratum_occurrence_specific_padstack_definition))
  SUBTYPE OF (ee_product_definition);
  reference_implementation : reference_packaged_part_interconnect_implementation;
  teardrop_taper_start_distance : OPTIONAL length_data_element;
  teardrop_taper_end_distance : OPTIONAL length_data_element;
  snowball_start_distance : OPTIONAL length_data_element;
  snowball_end_distance : OPTIONAL length_data_element;
  snowball_initial_spacing : OPTIONAL length_data_element;
  snowball_end_spacing : OPTIONAL length_data_element;
  snowball_initial_radius : OPTIONAL length_data_element;
  snowball_end_radius : OPTIONAL length_data_element;
  snowball_quantity_per_trace : OPTIONAL positive_integer;
INVERSE
  shape : padstack_definition_shape FOR shape_characterized_padstack_definition;
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*&#62;
    USEDIN(SELF\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') |
      (prpc.name = 'template model')
    )) &#62;= 0;
END_ENTITY; -- padstack_definition
( *
```

4.2.596 Padstack_definition_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Padstack_definition_shape is a type of Managed_design_object (see 4.2.524). A Padstack_definition_shape is the planar shape of a Part_template (see 4.2.620).

The data associated with a Padstack_definition_shape are the following:

- shape_characterized_padstack_definition;
- shape_environment;
- shape_material_condition;
- shape_representation.

4.2.596.1 shape_characterized_padstack_definition

The `shape_characterized_padstack_definition` specifies the role of the `Part_template` (see 4.2.620) for the `Padstack_definition_shape`.

4.2.596.2 shape_environment

The `shape_environment` specifies either manufacturing or an `end_user_application` as the valid environment for the `Padstack_definition_shape`. The `shape_environment` need not be specified for a particular `Padstack_definition_shape`.

4.2.596.3 shape_material_condition

The `shape_material_condition` specifies either the `maximum_material_condition`, the `minimum_material_condition`, or the `nominal_material_condition` as the material condition for the `Padstack_definition_shape`.

4.2.596.4 shape_representation

The `shape_representation` specifies the `Planar_projected_shape` (see 4.2.677) that is the model for the `Padstack_definition_shape`.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_definition_shape
  SUBTYPE OF (planar_projected_shape);
    shape_characterized_padstack_definition : padstack_definition;
    shape_environment                       : OPTIONAL environment;
    shape_material_condition                : material_condition;
END_ENTITY; -- padstack_definition_shape
( *
```

4.2.597 Padstack_location_in_footprint_definition

A `Padstack_location_in_footprint_definition` is a type of `Managed_design_object` (see 4.2.524).

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_location_in_footprint_definition
  SUBTYPE OF (managed_design_object);
    padstack : padstack_definition;
    footprint : footprint_definition;
    reference_package_terminal : package_terminal;
END_ENTITY; -- padstack_location_in_footprint_definition
( *

```

4.2.598 Padstack_location_in_footprint_definition_transform

A Padstack_location_in_footprint_definition_transform is a type of Managed_design_object (see 4.2.524).

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_location_in_footprint_definition_transform
  SUBTYPE OF (managed_design_object);
    padstack_shape : padstack_definition_shape;
    footprint_shape : footprint_definition_shape;
    reference_location : padstack_location_in_footprint_definition;
    transform : axis_placement_2d;
END_ENTITY; -- padstack_location_in_footprint_definition_transform
( *

```

4.2.599 Padstack_occurrence

A Padstack_occurrence is a type of Assembly_group_component (see 4.2.40). A Padstack_occurrence organizes the various Land and Inter_stratum_features so that one placement may suffice for all Application Objects that are derived from the placement of one padstack.

The data associated with a Padstack_occurrence are the following:

— usage_definition.

4.2.599.1 usage_definition

The usage_definition inherited from the Assembly_component specifies the role of the Padstack_definition (see 4.2.595) for the Padstack_occurrence.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_occurrence
  SUBTYPE OF (assembly_group_component);
  SELF\assembly_component.usage_definition : padstack_definition;
WHERE
  WR1: SIZEOF(TYPEOF(SELF) * [
    'AP210_ARM.PRINTED_COMPONENT',
    'AP210_ARM.LAMINATE_COMPONENT',
    'AP210_ARM.PHYSICAL_COMPONENT']) = 0;
END_ENTITY; -- padstack_occurrence
( *

```

4.2.600 Padstack_occurrence_sub_assembly_relationship

A Padstack_occurrence_sub_assembly_relationship is a type of Managed_design_object (see 4.2.524). A Padstack_occurrence_sub_assembly_relationship is an association that specifies a subassembly constraint among two or more Laminate_component (see 4.2.483) that is aggregated into an Padstack_occurrence (see 4.2.599). A Land may participate in a Padstack_occurrence. In that case, there is no need to populate a separate Assembly_composition_relationship between the Land and the Pcb, since the Assembly_composition_relationship between the Padstack_occurrence and the Pcb establishes the required association. Also, in that case, there is no need to separately populate the usage_definition from Land to Land_physical_template because the Land template may be deduced from the Padstack_definition structure. A Via may appear in a Padstack_occurrence.

The data associated with a Padstack_occurrence_sub_assembly_relationship are the following:

- sub_assembly_reference_designation;
- related_component;
- relating_component.

4.2.600.1 sub_assembly_reference_designation

The sub_assembly_reference_designation specifies the string which is the identifier for this Padstack_occurrence_sub_assembly_relationship in the context of the relating_component.

4.2.600.2 related_component

The related_component specifies the dependent Laminate_component (see 4.2.483) of the Padstack_occurrence_sub_assembly_relationship.

4.2.600.3 relating_component

The relating_component specifies the independent Padstack_occurrence (see 4.2.599) which is the aggregation of two or more Assembly_component (see 4.2.34) specified by related_component.

Constraints:

UR1: The combination of relating_component, resident_stratum of the related_component, and sub_assembly_reference_designation shall be unique in a population of Padstack_occurrence_sub_assembly_relationship.

WR1: An Assembly_component specified as a relating_component shall not appear in the chain of components in the set of Assembly_component specified by related_component.

NOTE EXPRESS specification:

```

*)
ENTITY padstack_occurrence_sub_assembly_relationship
  SUBTYPE OF (managed_design_object);
    related_component : laminate_component;
    relating_component : padstack_occurrence;
    sub_assembly_reference_designation : string;
UNIQUE
  UR1: sub_assembly_reference_designation,
        relating_component,
        related_component;
END_ENTITY; -- padstack_occurrence_sub_assembly_relationship
( *
```

4.2.601 Parallel_shape_element_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Parallel_shape_element_relationship is a type of Shape_element_relationship (see 4.2.816) that asserts that two Shape_element (see 4.2.810) are parallel to each other. One Shape_element is specified as the related_shape_element and the other Shape_element is specified as the relating_shape_element.

NOTE 1 The attribute related_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

NOTE 2 The attribute relating_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

Constraints:

WR1: The Shape_element specified as the relating_shape_element shall not be specified as parallel to itself either directly or through intermediate Parallel_shape_element_relationship.

NOTE EXPRESS specification:

```
* )
ENTITY parallel_shape_element_relationship
  SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- parallel_shape_element_relationship

( *
```

4.2.602 Parallelism_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Parallelism_tolerance is a type of Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that specifies or partially specifies the design requirement on the parallelism of a Shape_element (see 4.2.810) in relation to a Datum_plane (see 4.2.232) or Datum_axis (see 4.2.215).

NOTE Parallelism tolerance is defined in terms of tolerance zone definitions in 14.7 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Centre_axis, Centre_plane, Physical_feature_or_part_template, or Tangent_plane.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: There shall be at most two Datum in the Datum_system the definition of which is specified as the datum_system_definition.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

WR3: At least one Datum_plane or Datum_axis shall be in the Datum_system the definition of which is specified as the datum_system_definition.

WR4: If the toleranced_shape_element is a Centre_plane, Physical_feature_or_part_template, or Tangent_plane, the Parallelism_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

IP1: The form of the tolerated Shape_element shall be consistent with the form of a shape element for which the application of a parallelism tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- parallelism_tolerance

( *
```

4.2.603 Parameter_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Parameter_assignment is a type of Managed_design_object (see 4.2.524). A Parameter_assignment is the association of either a Characteristic (see 4.2.98) or Coordinated_characteristic (see 4.2.197) with a Model_parameter (see 4.2.549), used to provide actual values for characteristics declared formally by the the Model_parameter.

The data associated with a Parameter_assignment are the following:

- assigned_parameter;
- parameter_value.

4.2.603.1 assigned_parameter

The assigned_parameter specifies the Model_parameter (see 4.2.549) that declares the formal parameters assigned values by the Parameter_assignment.

4.2.603.2 parameter_value

The parameter_value specifies either the role of a Characteristic (see 4.2.98) or a Coordinated_characteristic (see 4.2.197) for the Parameter_assignment.

Constraints:

WR1: The type of the parameter_value shall be the same as that of the assigned_parameter.

NOTE EXPRESS specification:

```

*)
ENTITY parameter_assignment
  SUBTYPE OF (managed_design_object);
    parameter_value : scalar_or_coordinated_characteristics;
    assigned_parameter : model_parameter;
END_ENTITY; -- parameter_assignment

--tc1 wd8 modification
( *
```

4.2.604 Part_array_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_array_feature is a type of Part_group_feature (see 4.2.613) that is an array of Part_feature (see 4.2.609) whose positions are specified as members of an array, and whose tolerances are applied based upon the array nature of Part_array_feature.

NOTE Tolerances applied to Part_array_feature are specified by an object external to the Part_array_feature.

NOTE EXPRESS specification:

```

*)
ENTITY part_array_feature
  ABSTRACT SUPERTYPE OF (ONEOF (part_linear_array_feature,
    part_rectangular_array_feature))
  SUBTYPE OF (part_group_feature);
END_ENTITY; -- part_array_feature

( *
```

4.2.605 Part_connected_terminals_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_connected_terminals_definition is a type of Managed_design_object (see 4.2.524). A Part_connected_terminals_definition is the association between two or more Part_terminal (see 4.2.630) that are connected internally to the part. This object serves as a formal requirement statement for connections implemented by a product. It is a topological abstraction in that it does not provide any information stating the variation of properties with respect to product or environmental variability.

NOTE This object is included to support the case where multiple terminals of a part are connected internally (e.g VCC, Ground) to facilitate mapping between the physical terminal definition and the functional terminal definition.

The data associated with a `Part_connected_terminals_definition` are the following:

- `associated_definition`;
- `connected_terminals`;
- `name`.

4.2.605.1 associated_definition

The `associated_definition` specifies the `Physical_unit_usage_view` (see 4.2.672) that is the product definition *s* for the `Part_connected_terminals_definition`.

4.2.605.2 connected_terminals

The `connected_terminals` specifies the role of the `Part_terminal` (see 4.2.630) for the `Part_connected_terminals_definition`. There shall be two or more `Part_terminal` for a `Part_connected_terminals_definition`.

4.2.605.3 name

The `name` identifies the `Part_connected_terminals_definition`.

Constraints:

UR1: The combination of the `name` and `associated_definition` shall be unique within a population of `Part_connected_terminals_definitions`.

WR1: All terminals specified by the member of this object shall be on the `associated_definition`.

WR2: All terminals specified by the member of this object shall be mapped to the same terminal of the functional definition.

NOTE EXPRESS specification:

```

*)
ENTITY part_connected_terminals_definition
  SUBTYPE OF (managed_design_object);
    associated_definition : physical_unit_usage_view;
    connected_terminals   : SET [2:?] OF part_terminal;
    name                  : ee_name;

  UNIQUE
    UR1 : name, associated_definition;
WHERE
  WR1 : SIZEOF(QUERY( ct &#60;* connected_terminals |
    NOT(associated_definition = ct\part_feature.associated_definition)
  )) = 0;
END_ENTITY; -- part_connected_terminals_definition

( *

```

4.2.606 Part_device

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_device is a type of Managed_design_object (see 4.2.524). A Part_device is a specific occurrence of a Bare_die (see 4.2.59) in a Packaged_part (see 4.2.591).

The data associated with a Part_device are the following:

- bonding_agent;
- defined_packaged_part;
- device_2d_position;
- device_3d_position;
- included_device;
- item_identification.

4.2.606.1 bonding_agent

The bonding_agent specifies the Ee_material (see 4.2.318) used to attach the Bare_die (see 4.2.59) within the Packaged_part (see 4.2.591). The bonding_agent need not be specified for a particular Part_device.

4.2.606.2 defined_packaged_part

The defined_packaged_part specifies the Packaged_part (see 4.2.591) that contains this Part_device.

4.2.606.3 device_2d_position

The device_2d_position is an inverse relationship that specifies the existence of the Device_2d_position (see 4.2.283) that specifies the Part_device as its placed_device. There shall be no more than one Device_2d_position for a Part_device.

4.2.606.4 device_3d_position

The device_3d_position is an inverse relationship that specifies the existence of the Device_3d_position (see 4.2.284) that specifies the Part_device as its placed_device. There shall be no more than one Device_3d_position for a Part_device.

4.2.606.5 included_device

The included_device specifies the Bare_die (see 4.2.59) used to provide the function of the Part_device.

4.2.606.6 item_identification

The item_identification specifies the name for the reference designation of the Bare_die instance within the Packaged_part.

Constraints:

UR1: The combination of the defined_package_part and the item_identification shall be unique within a population of Part_devices.

NOTE EXPRESS specification:

```

*)
ENTITY part_device
  SUBTYPE OF (managed_design_object);
  defined_package_part : packaged_part;
  included_device      : bare_die;
  item_identification  : ee_name;
  bonding_agent       : OPTIONAL ee_material;
  INVERSE
    device_3d_position : SET [0:1] OF device_3d_position FOR placed_device;
    device_2d_position : SET [0:1] OF device_2d_position FOR placed_device;
  UNIQUE
    UR1 : defined_package_part, item_identification;
END_ENTITY; -- part_device

( *
```

4.2.607 Part_device_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_device_terminal is a type of Managed_design_object (see 4.2.524). A Part_device_terminal is the surface area on an occurrence of a Bare_die (see 4.2.59) in a Packaged_part (see 4.2.591) that shall allow access to the function of an occurrence of the Bare_die. A Part_device_terminal is included to support detailed modeling and analysis of the internal connection to a Packaged_part_terminal. No connection between Part_device_terminals is supported in this part of ISO 10303. Connections between Bare_die are supported only within Assembly_module.

The data associated with a Part_device_terminal are the following:

- terminated_device;
- used_bare_die_terminal.

4.2.607.1 terminated_device

The terminated_device specifies the Part_device (see 4.2.606) that has this Part_device_terminal.

4.2.607.2 used_bare_die_terminal

The used_bare_die_terminal specifies the Bare_die_terminal (see 4.2.66) that shall be used by the Part_device_terminal.

Constraints:

UR1: The combination of the terminated_device and the used_bare_die_terminal shall be unique within a population of Part_device_terminals.

NOTE EXPRESS specification:

```
*)
ENTITY part_device_terminal
  SUBTYPE OF (managed_design_object);
    used_bare_die_terminal : minimally_defined_bare_die_terminal;
    terminated_device      : part_device;
  UNIQUE
    UR1 : used_bare_die_terminal, terminated_device;
END_ENTITY; -- part_device_terminal
```

(*

4.2.608 Part_device_terminal_model_port_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_device_terminal_model_port_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Part_device_terminal_model_port_assignment` is the relationship between the `Analytical_model_port` (see 4.2.17) and the `Part_device_terminal` (see 4.2.607), for the purpose of supporting behavioural simulation of bond wire and lead frame effects in a full package simulation. A simulation model provides an explicitly defined functional block that acts as the interface between the `Part_device_terminals` that are the access mechanisms of the `Part_device` and the `Packaged_part_terminals` that are the access mechanisms of the `Packaged_part`. This is appropriate where there is no detailed physical information available. The purpose, fidelity, and accuracy of the simulation model are not predefined in this part of ISO 10303.

NOTE 1 The `Device_terminal_map` (see 4.2.285) is the mechanism provided in this part of ISO 10303 for defining the "internal" surface of the `Package_terminal` (see 4.2.581) as used in a `Packaged_part` (see 4.2.591) since the design of a `Packaged_part` is outside the scope of this part of ISO 10303.

NOTE 2 Various manufacturers provide simulation models for the combination of leadframe and IC bond whose internal connection may be referenced by this object and whose external connection may be assigned to the `Package_terminal` by the `Part_feature_model_port_assignment` (see 4.2.611) object.

NOTE 3 This part of ISO 10303 does not specify the physical details of the connection from the `Package_terminal` to the `Part_device_terminal` since that is outside the scope of the `Packaged_part` Application Object.

NOTE 4 A single IC package design definition (e.g. Ball-grid design definition) may be exchanged with this part of ISO 10303 by considering the package to be an `Interconnect_module` (see 4.2.440).

The data associated with a `Part_device_terminal_model_port_assignment` are the following:

- `internal_connection`;
- `internal_port`.

4.2.608.1 internal_connection

The `internal_connection` specifies the role of the `Part_device_terminal` (see 4.2.607) for the `Part_device_terminal_model_port_assignment`.

4.2.608.2 internal_port

The `internal_port` specifies the role of the `Analytical_model_port` (see 4.2.17) for the `Part_device_terminal_model_port_assignment`.

Constraints:

UR1: The combination of the `internal_connection` and the `internal_port` shall be unique within a population of `Part_device_terminal_model_port_assignment`.

NOTE EXPRESS specification:

```

*)
ENTITY part_device_terminal_model_port_assignment
  SUBTYPE OF (managed_design_object);
    internal_connection : part_device_terminal;
    internal_port       : analytical_model_port;
  UNIQUE
    UR1 : internal_connection, internal_port;
END_ENTITY; -- part_device_terminal_model_port_assignment

( *

```

4.2.609 Part_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_feature is a type of Physical_feature_or_part_template (see 4.2.658). A Part_feature may be one or more of a Part_group_feature (see 4.2.613), a Part_mounting_feature (see 4.2.618), a Part_tooling_feature (see 4.2.633), a Thermal_feature (see 4.2.901), a Package_terminal (see 4.2.581), or a Part_terminal (see 4.2.630); or additionally one of a Primary_orientation_feature (see 4.2.717), or a Package_body (see 4.2.574), or an Interconnect_module_surface_feature (see 4.2.462).

NOTE 1 A Part_feature is a distinctive aspect of a Physical_unit (see 4.2.664) that partially supports the choice of that Physical_unit in the design.

NOTE 2 The design_view_definition_component_feature and design_view_definition_stratum_concept should only be populated for design archival and should not normally be publicly available as they allow queries against detailed design data.

The data associated with a Part_feature are the following:

- associated_definition;
- design_view_definition_component_feature;
- design_view_definition_stratum_concept;
- feature_shape;
- name;
- precedent_feature.

4.2.609.1 associated_definition

The associated_definition specifies the Physical_unit_usage_view (see 4.2.672) for the targeted Part_feature.

4.2.609.2 design_view_definition_component_feature

The design_view_definition_component_feature is an inverse relationship that specifies the Component_feature_to_physical_usage_view_assignment (see 4.2.123) that specifies the Part_feature as its defined_part_feature. There shall be no more than one Component_feature_to_physical_usage_view_assignment for a Part_feature.

4.2.609.3 design_view_definition_stratum_concept

The design_view_definition_stratum_concept is an inverse relationship that specifies the Stratum_concept_to_physical_usage_view_assignment (see 4.2.850) that specifies the Part_feature as its defined_part_feature. There shall be no more than one Stratum_concept_to_physical_usage_view_assignment for a Part_feature.

4.2.609.4 feature_shape

The feature_shape specifies the role of the Feature_shape_occurrence (see ??) for the Part_feature. The feature_shape need not be specified for a particular Part_feature. There may be more than one Feature_shape_occurrence for a Part_feature.

4.2.609.5 name

The name specifies the descriptive label recognizable within the electronics community and by a computer system for the Part_feature. The name is a redeclaration of the name attribute inherited from Shape_element and shall be interpreted as the name attribute of shape_aspect.

4.2.609.6 precedent_feature

The precedent_features specifies the role of the Part_feature for the Part_feature. The precedent_feature need not be specified for a particular Part_feature.

Constraints:

UR1: The combination of the associated_definition and the name shall be unique within a population of Part_features.

WR1: The shape of features shall be consistent with the shape of the physical_unit_usage_view defined by associated_definition.

WR2: The Part_feature shall not participate in a cyclic relationship with the precedent_feature, either directly or indirectly.

WR3: Each Part_feature shall not be assigned by both a Stratum_concept_to_physical_usage_view_assignment and a Component_feature_to_physical_usage_view_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY part_feature
  SUPERTYPE OF ((((((((((part_group_feature ANDOR
    ONEOF(primary_orientation_feature,
      secondary_orientation_feature,
      tertiary_orientation_feature,
      package_body))) ANDOR
      part_tooling_feature) ANDOR
      thermal_feature) ANDOR
      part_interface_access_feature) ANDOR
      part_mating_feature) ANDOR
      part_mounting_feature) ANDOR
      part_terminal) ANDOR
      interconnect_module_surface_feature) ANDOR
      package_terminal) ANDOR
      package_body_surface) ANDOR
      polarity_indication_feature)
  SUBTYPE OF (physical_feature_or_part_template);
  associated_definition      : physical_unit_usage_view;
  material_state_change     : OPTIONAL material_state_change_enumeration;
  precedent_feature         : OPTIONAL part_feature;
  SELF\shape_element.name : ee_name;
INVERSE
  design_view_definition_stratum_concept :
SET [0:1] OF stratum_concept_to_physical_usage_view_assignment FOR
defined_part_feature;
  design_view_definition_component_feature :
SET [0:1] OF component_feature_to_physical_usage_view_assignment FOR
defined_part_feature;
  feature_shape              : SET [0:?] OF usage_concept_usage_relationship FOR
  associated_usage;
UNIQUE
  UR1 : SELF\shape_element.name, associated_definition;
END_ENTITY; -- part_feature

( *
```

4.2.610 Part_feature_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_feature_make_from_relationship is a type of Managed_design_object (see 4.2.524).

The data associated with a Part_feature_make_from_relationship are the following:

- associated_make_from;
- reusable_feature;
- resultant_feature.

4.2.610.1 associated_make_from

The associated_make_from specifies an Engineering_make_from (see 4.2.353) for the Part_feature_make_from_relationship.

4.2.610.2 reusable_feature

The reusable_feature specifies a Part_feature (see 4.2.609) for the Part_feature_make_from_relationship.

4.2.610.3 resultant_feature

The resultant_feature specifies a Part_feature (see 4.2.609) for the Part_feature_make_from_relationship.

Constraints:

UR1: The combination of reusable_feature and resultant_feature shall be unique within a population of Part_feature_make_from_relationship.

WR1: The reusable_feature shall not be the resultant_feature within a population of Part_feature_make_from_relationship.

WR2: The Physical_unit_usage_view referenced by the reusable_feature shall not be the Physical_unit_usage_view referenced by the resultant_feature within a population of Part_feature_make_from_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY part_feature_make_from_relationship
  SUBTYPE OF (managed_design_object);
    associated_make_from : engineering_make_from;
    reusable_feature : part_feature;
    resultant_feature : part_feature;
UNIQUE
  UR1: reusable_feature, resultant_feature;
WHERE
  WR1: reusable_feature :&#60;&#62;; resultant_feature;
  WR2: reusable_feature.associated_definition :&#60;&#62;;
    resultant_feature.associated_definition;
END_ENTITY; -- part_feature_make_from_relationship

( *
```

4.2.611 Part_feature_model_port_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_feature_model_port_assignment is a type of Managed_design_object (see 4.2.524). A Part_feature_model_port_assignment is the relationship between an Analytical_model_port (see 4.2.17) and the Part_feature (see 4.2.609).

NOTE Usually a simulation model is associated on a one to one basis with the Physical_unit_usage_view.

The data associated with a Part_feature_model_port_assignment are the following:

- source;
- target.

4.2.611.1 source

The source specifies the Part_feature (see 4.2.609) whose port is to be associated with an Analytical_model_port (see 4.2.17) in a Part_feature_model_port_assignment.

4.2.611.2 target

The target specifies the Analytical_model_port (see 4.2.17) that is to be associated with a Part_feature (see 4.2.609) in a Part_feature_model_port_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY part_feature_model_port_assignment
  SUBTYPE OF (managed_design_object);
    source : part_feature;
    target : analytical_model_port;
END_ENTITY; -- part_feature_model_port_assignment

--tt tc1wd8 addition
( *

```

4.2.612 Part_feature_placement_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_feature_placement_link is a type of a Managed_design_object (see 4.2.524).

The data associated with a Part_feature_placement_link are the following:

- precedent_element;
- referenced_by;
- subsequent_element.

4.2.612.1 precedent_element

The precedent_element specifies a Part_feature (see 4.2.609) for the Part_feature_placement_link.

4.2.612.2 referenced_by

The referenced_by is an inverse relationship that specifies that the existence of the Part_feature_placement_link is dependent on the existence of the Part_linear_array_feature (see 4.2.615) that specifies the Part_feature_placement_link as its element.

4.2.612.3 subsequent_element

The subsequent_element specifies a Part_feature (see 4.2.609) for the Part_feature_placement_link.

NOTE EXPRESS specification:

```
*)
ENTITY part_feature_placement_link
  SUBTYPE OF (managed_design_object);
  precedent_element : part_feature;
  subsequent_element : part_feature;
INVERSE
  referenced_by : part_linear_array_feature FOR
    element;
END_ENTITY; -- part_feature_placement_link

( *
```

4.2.613 Part_group_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_group_feature is a type of Part_feature (see 4.2.609) that is an aggregation of Part_features. A Part_group_feature may be a Part_array_feature (see 4.2.604).

NOTE EXPRESS specification:

```
*)
ENTITY part_group_feature
  SUPERTYPE OF (part_array_feature)
  SUBTYPE OF (part_feature);
WHERE
  WR1:SIZEOF(QUERY( fgr &#60;* USEDIN(SELF,
    'AP210_ARM.FEATURE_GROUP_RELATIONSHIP.RELATING_FEATURE') |
    fgr.related_feature\part_feature.associated_definition &#60;&#62;
    SELF\part_feature.associated_definition)) = 0;
END_ENTITY; -- part_group_feature

( *
```

4.2.614 Part_interface_access_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_interface_access_feature is a type of Part_feature (see 4.2.609) that is identified to facilitate tolerance specification when an occurrence of the associated part is used in an assembly. Specifically the intent is to provide identification that the function of the Part_feature is to provide access via a void in the material to internal components or terminals, which have been identified as having some function at the Part level, e.g., are Part_terminals. The material_state_change attribute inherited from Part_feature shall be material_removal.

EXAMPLE 1 The reset button in a notebook computer requires a Part_interface_access_feature whose properties are coupled to the reset button to ensure correct functionality of the end product.

The data associated with a Part_interface_access_feature are the following:

- accessed_interface;
- interface_access_connection_area.

4.2.614.1 accessed_interface

The accessed_interface specifies the role of the Part_feature (see 4.2.609) for the Part_interface_access_feature. The accessed_interface is the feature for which the Part_interface_access_feature provides access.

NOTE The accessed_interface is at the same level of design hierarchy as the Part_interface_access_feature for consistency purposes. It is anticipated that component_feature_to_physical_usage_view_assignment will be populated as necessary to support the design process peculiar to the enterprise.

4.2.614.2 interface_access_connection_area

The interface_access_connection_area specifies the role of the Connection_zone (see 4.2.185) for the Part_interface_access_feature. The accessed_interface specifies the role of the Part_feature (see 4.2.609) for the Part_interface_access_feature. The interface_access_connection_area provides a means to geometrically specify an area for critical mechanical interface_access properties. The Connection_zone area is void of material.

NOTE EXPRESS specification:

```

*)
ENTITY part_interface_access_feature
  SUBTYPE OF (part_feature);
  accessed_interface : part_feature;
  interface_access_area : OPTIONAL SET [1:?] OF connection_zone;
WHERE
  WR1: SELF\part_feature.material_state_change = material_removal;
  WR2: SELF &#60;&#62; accessed_interface;
  WR3: NOT('AP210_ARM.PART_TERMINAL' IN TYPEOF(SELF));
  WR4: NOT('AP210_ARM.PACKAGE_BODY' IN TYPEOF(SELF));
  WR5: NOT('AP210_ARM.PART_MOUNTING_FEATURE' IN TYPEOF(SELF));
  WR6: NOT('AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE' IN TYPEOF(SELF));
  WR7: NOT('AP210_ARM.PACKAGE_TERMINAL' IN TYPEOF(SELF));
  WR8: NOT('AP210_ARM.PACKAGE_BODY_SURFACE' IN TYPEOF(SELF));
  WR9: NOT('AP210_ARM.POLARITY_INDICATION_FEATURE' IN TYPEOF(SELF));
END_ENTITY; -- part_interface_access_feature

( *
```

4.2.615 Part_linear_array_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_linear_array_feature is a type of Part_array_feature (see 4.2.604). The location_uncertainty refers to originating CAD system geometric uncertainty, not material processing tolerance.

The data associated with a Part_linear_array_feature are the following:

- element;
- initial_node_positive_x_displacement;
- terminus_node_positive_x_displacement.

4.2.615.1 element

The element specifies the role of a Part_feature_placement_link (see 4.2.612) for the Part_linear_array_feature. There shall be one or more Part_feature_placement_link for a Part_linear_array_feature.

4.2.615.2 initial_node_positive_x_displacement

The initial_node_positive_x_displacement specifies a Length_data_element (see 4.2.511) for the Part_linear_array_feature.

4.2.615.3 terminus_node_positive_x_displacement

The terminus_node_positive_x_displacement specifies a Length_data_element (see 4.2.511) for the Part_linear_array_feature.

NOTE EXPRESS specification:

```

*)
ENTITY part_linear_array_feature
  SUBTYPE OF (part_array_feature);
  element : SET [1:?] OF part_feature_placement_link;
  initial_node_positive_x_displacement : length_data_element;
  terminus_node_positive_x_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- part_linear_array_feature
(

```

4.2.616 Part_linear_array_feature_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_linear_array_feature_link` is a type of a `Managed_design_object` (see 4.2.524).

The data associated with a `Part_linear_array_feature_link` are the following:

- `precedent_element`;
- `referenced_by`;
- `subsequent_element`.

4.2.616.1 `precedent_element`

The `precedent_element` specifies a `Part_linear_array_feature` (see 4.2.615) for the `Part_linear_array_feature_link`.

4.2.616.2 `referenced_by`

The `referenced_by` is an inverse relationship that specifies that the existence of the `Part_linear_array_feature_link` is dependent on the existence of the `Part_rectangular_array_feature` (see 4.2.619) that specifies the `Part_linear_array_feature_link` as its element.

4.2.616.3 `subsequent_element`

The `subsequent_element` specifies a `Part_linear_array_feature` (see 4.2.615) for the `Part_linear_array_feature_link`.

NOTE EXPRESS specification:

```

*)
ENTITY part_linear_array_feature_link
  SUBTYPE OF (managed_design_object);
    precedent_element : part_linear_array_feature;
    subsequent_element : part_linear_array_feature;
INVERSE
  referenced_by : part_rectangular_array_feature FOR
                  element;
END_ENTITY; -- part_linear_array_feature_link

--tt tc1wd9 modified supertype constraint
--tt tc1wd8 added supertype constraint
( *
```

4.2.617 `Part_mating_feature`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_mating_feature` is a type of `Part_feature` (see 4.2.609) that is identified to facilitate tolerance specification when an occurrence of the associated part is mated to a complementary part. A `Part_mating_feature` is provided where there is no requirement to convey electrical functionality as would be provided by `Part_terminal` (see 4.2.630).

NOTE The use of multi-pin connectors may require specification of mechanical mating areas in addition to electrical terminals.

EXAMPLE 1 The plastic housing on some connectors may contain `Part_mating_features`.

The data associated with a `Part_mating_feature` are the following:

— `mating_connection_area`.

4.2.617.1 `mating_connection_area`

The `mating_connection_area` specifies the role of the `Connection_zone` (see 4.2.185) for the `Part_mating_feature`. The `mating_connection_area` provides a means to geometrically specify a surface area for critical mechanical mating properties.

NOTE The `Feature_shape_occurrence` (see ??) specified by the mating feature may be the same `Feature_shape_occurrence` that is specified by the associated `Connection_zone` (see 4.2.185). In this case the whole mating feature is the `Connection_zone`.

NOTE EXPRESS specification:

```
*)
ENTITY part_mating_feature
  SUBTYPE OF (part_feature);
    mating_area : OPTIONAL SET [1:?] OF connection_zone;
WHERE
  WR1: NOT('AP210_ARM.PART_MOUNTING_FEATURE' IN TYPEOF(SELF));
  WR2: NOT('AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE' IN TYPEOF(SELF));
  WR3: NOT('AP210_ARM.PACKAGE_TERMINAL' IN TYPEOF(SELF));
  WR4: NOT('AP210_ARM.PART_TERMINAL' IN TYPEOF(SELF));
END_ENTITY; -- part_mating_feature

( *
```

4.2.618 `Part_mounting_feature`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_mounting_feature` is a type of `Part_feature` (see 4.2.609) that is identified to facilitate tolerance specification when an occurrence of the associated part is used in an assembly.

NOTE The use of high density surface mount parts may require specification of mechanical mounting areas in addition to electrical terminals.

The data associated with a `Part_mounting_feature` are the following:

— `mounting_connection_area`.

4.2.618.1 `mounting_connection_area`

The `mounting_connection_area` specifies the role of the `Connection_zone` (see 4.2.185) for the `Part_mounting_feature`. The `mounting_connection_area` provides a means to geometrically specify a surface area for critical mechanical mounting properties.

NOTE The `Feature_shape_occurrence` (see ??) specified by the mounting feature may be the same `Feature_shape_occurrence` that is specified by the associated `Connection_zone` (see 4.2.185). In this case the whole mounting feature is the `Connection_zone`.

NOTE EXPRESS specification:

```
*)
ENTITY part_mounting_feature
  SUPERTYPE OF (bare_die_surface)
  SUBTYPE OF (part_feature);
  mounting_area : OPTIONAL SET [1:?] OF connection_zone;
END_ENTITY; -- part_mounting_feature

--tt tc1 wd8 addition
(*
```

4.2.619 `Part_rectangular_array_feature`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_rectangular_array_feature` is a type of `Part_array_feature` (see 4.2.604). The `location_uncertainty` refers to originating CAD system uncertainty, not material processing tolerances.

The data associated with a `Part_rectangular_array_feature` are the following:

— `element`;

— `initial_row_positive_y_displacement`;

— `terminus_row_positive_y_displacement`.

4.2.619.1 element

The element specifies the role of a Part_linear_array_feature_link (see 4.2.616) for the Part_rectangular_array_feature. There shall be one or more Part_linear_array_feature_link for a Part_rectangular_array_feature.

4.2.619.2 initial_row_positive_y_displacement

The initial_row_positive_y_displacement specifies a Length_data_element (see 4.2.511) for the Part_rectangular_array_feature.

4.2.619.3 terminus_row_positive_y_displacement

The terminus_row_positive_y_displacement specifies a Length_data_element (see 4.2.511) for the Part_rectangular_array_feature.

NOTE EXPRESS specification:

```

*)
ENTITY part_rectangular_array_feature
  SUBTYPE OF (part_array_feature);
  element : SET [1:?] OF part_linear_array_feature_link;
  initial_row_positive_y_displacement : length_data_element;
  terminus_row_positive_y_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- part_rectangular_array_feature
( *
```

4.2.620 Part_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template is a type of Physical_feature_or_part_template (see 4.2.658). A Part_template may be one of an Inter_stratum_feature_template (see 4.2.435), a Material_addition_feature_template (see 4.2.529), a Material_removal_feature_template (see 4.2.534), a Printed_part_template (see 4.2.730), a Special_symbol_part_template (see 4.2.835), a Stratum_feature_template (see 4.2.861), or a Trace_template (see 4.2.930), or additionally may be a Part_text_template (see 4.2.632). One reason for including information provided by members of Part_template is that they are collectors for shape data templates for layout items.

The data associated with a Part_template are the following:

- analytical_part_template_representation;
- part_template_analytical_model;
- physical_characteristic.

4.2.620.1 analytical_part_template_representation

The analytical_part_template_representation specifies the role of the Analytical_representation (see 4.2.22) for the Part_template. There may be more than one Analytical_representation for the Part_template. The analytical_part_template_representation need not be specified for a particular Part_template.

4.2.620.2 part_template_analytical_model

The part_template_analytical_model specifies the role of the Analytical_model (see 4.2.14) for the Part_template. The part_template_analytical_model need not be specified for a particular Part_template.

4.2.620.3 physical_characteristic

The physical_characteristic specifies either the Characteristic (see 4.2.98) or Coordinated_characteristic (see 4.2.197) for the Part_template. There may be more than one Characteristic for the Part_template. The physical_characteristic need not be specified for a particular Part_template.

NOTE EXPRESS specification:

```

*)
ENTITY part_template
  SUPERTYPE OF (ONEOF (
    inter_stratum_feature_template,
    material_addition_feature_template,
    material_removal_feature_template,
    non_conductive_cross_section_template,
    printed_part_cross_section_template,
    printed_part_template,
    special_symbol_part_template,
    stratum_feature_template,
    trace_template)
  ANDOR part_text_template)
  SUBTYPE OF (physical_feature_or_part_template);
  part_template_analytical_model      : OPTIONAL analytical_model;
  physical_characteristic              : OPTIONAL SET [1:?] OF
                                         correlated_or_independent;
  analytical_part_template_representation : OPTIONAL SET [1:?] OF
                                         analytical_representation;
  associated_definition               : template_definition;
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*
    USEDIN(associated_definition\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') |
      (prpc.name = 'template model')
    )) &#62;= 0;
  WR2: SELF\shape_element.name = 'NULL';
END_ENTITY; -- part_template
(

```

4.2.621 Part_template_3d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_3d_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Part_template_3d_shape is the shape in three dimensions of the Part_template (see 4.2.620).

The data associated with a Part_template_3d_shape are the following:

- shape_characterized_part_template;
- shape_environment;
- shape_material_condition;
- shape_representation.

4.2.621.1 shape_characterized_part_template

The shape_characterized_part_template specifies the role of the Part_template (see 4.2.620) for the Part_template_3d_shape.

4.2.621.2 shape_environment

The shape_environment specifies either manufacturing or the end_user_application as the environment in which the Part_template_3d_shape is valid. The shape_environment need not be specified for a particular Part_template_3d_shape.

4.2.621.3 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition or the nominal_material_condition as the material condition for the Part_template_3d_shape.

4.2.621.4 shape_representation

The shape_representation specifies the Bound_volume_shape (see 4.2.76) that is the model for the Part_template_3d_shape.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_3d_shape
--tt tc1 wd9 modified
    SUBTYPE OF (bound_volume_shape, feature_shape_definition);
        shape_material_condition      : material_condition;
        shape_environment              : OPTIONAL environment;
(* tt tc1 wd9 deleted
    shape_representation              : bound_volume_shape;
*)
    shape_characterized_part_template : part_template;
END_ENTITY; -- part_template_3d_shape

( *
```

4.2.622 Part_template_3d_keepout_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_3d_keepout_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Non_feature_shape_definition (see 4.2.559). A Part_template_3d_keepout_shape is the shape in three orthogonal dimensions of a keepout constraint for a Part_template (see 4.2.620).

NOTE Keepout regions typically include some minimal spacing around the extreme shape.

The data associated with a `Part_template_3d_keepout_shape` are the following:

- `application_technology_constraint`;
- `centroid_location`;
- `constrained_design_object_category`;
- `shape_characterized_part_template`;
- `shape_environment`;
- `shape_material_condition`;
- `shape_purpose`;
- `shape_representation`.

4.2.622.1 application_technology_constraint

The `application_technology_constraint` specifies the role of `Ee_requirement_occurrence` for the `Part_template_3d_keepout_shape`. The `application_technology_constraint` is the relationship between keepout shapes and process constraints under which those keepout shapes are valid.

EXAMPLE 1 An example of an `application_technology_constraint` is the relationship of the shape of a clearance area to an assembly or testing process which requires the clearance.

4.2.622.2 centroid_location

The `centroid_location` is a `Cartesian_point` (see 4.2.87) which is the center of a `Part_template_3d_keepout_shape`. A `centroid_location` need not be specified for a particular `Part_template_3d_keepout_shape`.

4.2.622.3 constrained_design_object_category

The `constrained_design_object_category` specifies the category for which the `Part_template_3d_keepout_shape` is a constraint.

4.2.622.4 shape_characterized_part_template

The `shape_characterized_part_template` specifies the role of the `Part_template` (see 4.2.620) for the `Part_template_3d_keepout_shape`.

4.2.622.5 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the environment for the Part_template_3d_keepout_shape.

4.2.622.6 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Part_template_3d_keepout_shape.

4.2.622.7 shape_purpose

The shape_purpose specifies either electro_magnetic_compatibility, shock, thermal, vibration, or an external_definition as the purpose for the Part_template_3d_keepout_shape.

4.2.622.8 shape_representation

The shape_representation specifies the role of the Bound_volume_shape (see 4.2.76) for the Part_template_3d_keepout_shape.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_3d_keepout_shape
  SUBTYPE OF (bound_volume_shape, non_feature_shape_definition);
  application_technology_constraint : OPTIONAL ee_requirement_occurrence;
  constrained_design_object_category : keepout_product_design_object_category;
  shape_characterized_part_template : part_template;
  shape_material_condition          : material_condition;
  shape_purpose                       : keepout_purpose;
  shape_environment                 : environment;
  centroid_location                 : OPTIONAL cartesian_point;
END_ENTITY; -- part_template_3d_keepout_shape

( *
```

4.2.623 Part_template_location_in_padstack_definition

A Part_template_location_in_padstack_definition is a type of Managed_design_object (see 4.2.524). A Part_template_location_in_padstack_definition is the relationship that identifies the location of a Part_template in a Padstack_definition. The Application Object references a Template_definition in the formal model since the interpretation of this Application Object is to an Assembly_component_usage, and since there is a one to one relationship between a Part_template and the associated Template_definition.

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_location_in_padstack_definition
  SUBTYPE OF (managed_design_object);
    padstack : padstack_definition;
    template : template_definition;
    reference_designation : string;
END_ENTITY; -- part_template_location_in_padstack_definition
( *

```

4.2.624 Part_template_location_in_padstack_definition_transform

A Part_template_location_in_padstack_definition_transform is a type of Managed_design_object (see 4.2.524).

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_location_in_padstack_definition_transform
  SUBTYPE OF (managed_design_object);
    padstack_shape : padstack_definition_shape;
    template_shape : part_template_planar_shape;
    reference_location : part_template_location_in_padstack_definition;
    transform : axis_placement_2d;
END_ENTITY; -- part_template_location_in_padstack_definition_transform
( *

```

4.2.625 Part_template_non_planar_2d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_non_planar_2d_shape is a type of Managed_design_object (see 4.2.524). A Part_template_non_planar_2d_shape is the shape of a Part_template (see 4.2.620) that is represented in the form of a surface.

The data associated with a `Part_template_non_planar_2d_shape` are the following:

- `shape_characterized_part_template`;
- `shape_environment`;
- `shape_material_condition`;
- `shape_representation`;
- `transformation`.

4.2.625.1 `shape_characterized_part_template`

The `shape_characterized_part_template` specifies the role of the `Part_template` (see 4.2.620) for the `Part_template_non_planar_2d_shape`.

4.2.625.2 `shape_environment`

The `shape_environment` specifies either manufacturing or an `end_user_application` as the valid environment for the `Part_template_non_planar_2d_shape`. The `shape_environment` need not be specified for a particular `Part_template_non_planar_2d_shape`.

4.2.625.3 `shape_material_condition`

The `shape_material_condition` specifies either the `maximum_material_condition`, the `minimum_material_condition`, or the `nominal_material_condition` as the material condition for the `Part_template_non_planar_2d_shape`.

4.2.625.4 `shape_representation`

The `shape_representation` specifies the `Open_shell_based_surface` (see 4.2.566) that is the model for the `Part_template_non_planar_2d_shape`.

4.2.625.5 `transformation`

The `transformation` specifies the role of the `Axis_placement_3d` (see 4.2.58) for the `Part_template_non_planar_2d_shape`.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_non_planar_2d_shape
  SUBTYPE OF (open_shell_based_surface, feature_shape_definition);
    transformation          : axis_placement_3d;
    shape_characterized_part_template : part_template;
    shape_environment        : OPTIONAL environment;
    shape_material_condition : material_condition;
END_ENTITY; -- part_template_non_planar_2d_shape
( *

```

4.2.626 Part_template_or_physical_unit_2d_shape_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_or_physical_unit_2d_shape_representation_relationship is a type of Managed_design_object (see 4.2.524).

The data associated with a Part_template_or_physical_unit_2d_shape_representation_relationship are the following:

- principal_part_shape;
- related_part_shape.

4.2.626.1 principal_part_shape

The principal_part_shape specifies the role of a Part_template_planar_shape (see 4.2.628) or a Physical_unit_planar_shape (see 4.2.670) for the Part_template_or_physical_unit_2d_shape_representation_relationship.

4.2.626.2 related_part_shape

The related_part_shape specifies the role of a Part_template_planar_shape (see 4.2.628) or a Physical_unit_planar_shape (see 4.2.670) for the Part_template_or_physical_unit_2d_shape_representation_relationship.

Constraints:

UR1: The combination of principal_part_shape and related_part_shape shall be unique within a population of Part_template_or_physical_unit_2d_shape_representation_relationship.

WR1: The principal_part_shape shall not be the related_part_shape within a population of Part_template_or_physical_unit_2d_shape_representation_relationship.

WR2: The `principal_part_shape` and `related_part_shape` shall be for the same definition or item and exist in the same `Cartesian_coordinate_system` (see 4.2.86) within a population of `Part_template_or_physical_unit_2d_shape_representation_relationship`.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_or_physical_unit_2d_shape_representation_relationship
  SUBTYPE OF (managed_design_object);
    principal_part_shape : part_template_or_physical_unit_2d_shape_select;
    related_part_shape   : part_template_or_physical_unit_2d_shape_select;
UNIQUE
  UR1: principal_part_shape, related_part_shape;
WHERE
  WR1: principal_part_shape :=&#60;&#62;; related_part_shape;
(
  WR2: principal_part_shape and related_part_shape shall be for the same definition
  or item and exist in the same Cartesian_coordinate_system.
*)
END_ENTITY; -- part_template_or_physical_unit_2d_-
            -- shape_representation_relationship

--tt tc1 correction added
(

```

4.2.627 Part_template_or_physical_unit_3d_shape_representation_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_template_or_physical_unit_3d_shape_representation_relationship` is a type of `Managed_design_object` (see 4.2.524).

The data associated with a `Part_template_or_physical_unit_3d_shape_representation_relationship` are the following:

- `principal_part_shape`;
- `related_part_shape`.

4.2.627.1 principal_part_shape

The `principal_part_shape` specifies the role of one of a `Part_template_3d_shape` (see 4.2.621), a `Part_template_non_planar_2d_shape` (see 4.2.625), or a `Physical_unit_3d_shape` (see 4.2.666) for the `Part_template_or_physical_unit_3d_shape_representation_relationship`.

4.2.627.2 related_part_shape

The related_part_shape specifies the role of one of a Part_template_3d_shape (see 4.2.621), a Part_template_non_planar_2d_shape (see 4.2.625), or a Physical_unit_3d_shape (see 4.2.666) for the Part_template_or_physical_unit_3d_shape_representation_relationship.

Constraints:

UR1: The combination of principal_part_shape and related_part_shape shall be unique within a population of Part_template_or_physical_unit_3d_shape_representation_relationship.

WR1: The principal_part_shape shall not be the related_part_shape within a population of Part_template_or_physical_unit_3d_shape_representation_relationship.

WR2: The principal_part_shape and related_part_shape shall be for the same definition or item and shall exist in the same Cartesian_coordinate_system (see 4.2.86) within a population of Part_template_or_physical_unit_3d_shape_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_or_physical_unit_3d_shape_representation_relationship
  SUBTYPE OF (managed_design_object);
    principal_part_shape : part_template_or_physical_unit_3d_shape_select;
    related_part_shape   : part_template_or_physical_unit_3d_shape_select;
UNIQUE
  UR1: principal_part_shape, related_part_shape;
WHERE
  WR1: principal_part_shape :&#60;&#62;;: related_part_shape;
( *
  WR2: principal_part_shape and related_part_shape shall be for the same definition
  or item and shall exist in the same Cartesian_coordinate_system.
*)
END_ENTITY; -- part_template_or_physical_unit_3d_-
            -- shape_representation_relationship

( *
```

4.2.628 Part_template_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_planar_shape is a type of Managed_design_object (see 4.2.524). A Part_template_planar_shape is the planar shape of a Part_template (see 4.2.620).

The data associated with a Part_template_planar_shape are the following:

- shape_characterized_part_template;
- shape_environment;
- shape_material_condition;
- shape_representation.

4.2.628.1 shape_characterized_part_template

The shape_characterized_part_template specifies the role of the Part_template (see 4.2.620) for the Part_template_planar_shape.

4.2.628.2 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the valid environment for the Part_template_planar_shape. The shape_environment need not be specified for a particular Part_template_planar_shape.

4.2.628.3 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Part_template_planar_shape.

4.2.628.4 shape_representation

The shape_representation specifies the Planar_projected_shape (see 4.2.677) that is the model for the Part_template_planar_shape.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_planar_shape
--tt tc1 wd9 modified
    SUBTYPE OF (planar_projected_shape);
    shape_characterized_part_template : part_template;
(* tt tc1 wd9 deleted
    shape_representation                : planar_projected_shape;
*)
    shape_environment                    : OPTIONAL environment;
    shape_material_condition              : material_condition;
END_ENTITY; -- part_template_planar_shape

( *
```

4.2.629 Part_template_planar_keepout_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_template_planar_keepout_shape is a type of Planar_shape (see 4.2.678) and a type of Non_feature_shape_definition (see 4.2.559). A Part_template_planar_keepout_shape is the projected shape of a keepout region around the Part_template (see 4.2.620) in two dimensions.

NOTE 1 Keepout regions typically include some minimal spacing around the extreme shape represented by Part_template_planar_shape where the exact relationship between the two shapes is unrecorded.

NOTE 2 There may be multiple Part_template_planar_keepout_shape for a particular purpose. The recommended method to aggregate several Part_template_planar_keepout_shape that together describe a volumetric constraint is to reference them from a member of Shape_and_ee_product_definition_based_constraint_occurrence (see 4.2.806).

The data associated with a Part_template_planar_keepout_shape are the following:

- application_technology_constraint;
- centroid_location;
- constrained_design_object_category;
- shape_characterized_part_template;
- shape_environment;
- shape_material_condition;
- shape_purpose.

4.2.629.1 application_technology_constraint

The application_technology_constraint specifies the role of Ee_requirement_occurrence for the Part_template_3d_shape. The application_technology_constraint is the relationship between keepout shapes and process constraints under which those keepout shapes are valid.

EXAMPLE 1 An example of an application_technology_constraint is the relationship of the shape of a clearance area to an assembly or testing process which requires the clearance.

4.2.629.2 centroid_location

The centroid_location is a Cartesian_point (see 4.2.87) which is the center of a Part_template_planar_keepout_shape. A centroid_location need not be specified for a particular Part_template_planar_keepout_shape.

4.2.629.3 constrained_design_object_category

The constrained_design_object_category specifies the category of design object for which the Part_template_planar_keepout_shape is a constraint.

4.2.629.4 shape_characterized_part_template

The shape_characterized_part_unit specifies the role of the Part_template (see 4.2.620) for the Part_template_planar_keepout_shape.

4.2.629.5 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the environment for the Part_template_planar_keepout_shape.

4.2.629.6 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Part_template_planar_keepout_shape.

4.2.629.7 shape_purpose

The shape_purpose specifies either electro_magnetic_compatibility, shock, thermal, vibration, or an external_definition as the purpose for the Part_template_planar_keepout_shape.

NOTE EXPRESS specification:

```

*)
ENTITY part_template_planar_keepout_shape
  SUBTYPE OF (planar_shape, non_feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    constrained_design_object_category : keepout_product_design_object_category;
    shape_characterized_part_template : part_template;
    shape_material_condition          : material_condition;
    centroid_location                 : OPTIONAL cartesian_point;
    shape_environment                 : environment;
    shape_purpose                      : keepout_purpose;
END_ENTITY; -- part_template_planar_keepout_shape

( *
```

4.2.630 Part_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_terminal is a type of Part_feature (see 4.2.609). A Part_terminal is one of an Assembly_module_terminal (see 4.2.50), a Bare_die_terminal (see 4.2.66), an Interconnect_module_terminal (see 4.2.463), or a Packaged_part_terminal (see 4.2.594).

The data associated with a Part_terminal are the following:

- connection_requirement;
- functional_usage_view_terminal.

4.2.630.1 connection_requirement

The connection_requirement is an inverse relationship that specifies the Part_connected_terminals_definition (see 4.2.605) that specifies the Part_terminal as its connected_terminals. There shall be no more than one Part_connected_terminals_definition for a Part_terminal.

4.2.630.2 functional_usage_view_terminal

The functional_usage_view_terminal is an inverse relationship to the Functional_usage_view_to_part_terminal_assignment (see 4.2.413) that specifies the Part_terminal as its physical_usage_view_terminal. There shall be no more than one Functional_usage_view_to_part_terminal_assignment (see 4.2.413) for a Part_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY part_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (minimally_defined_bare_die_terminal,
    assembly_module_terminal, interconnect_module_terminal,
    packaged_part_terminal))
  SUBTYPE OF (part_feature);
INVERSE
  connection_requirement      : SET [0:1] OF
part_connected_terminals_definition FOR connected_terminals;
  functional_usage_view_terminal :
SET [0:1] OF functional_usage_view_to_part_terminal_assignment FOR
physical_usage_view_terminal;
END_ENTITY; -- part_terminal

( *
```

4.2.631 Part_terminal_external_reference

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Part_terminal_external_reference` is a type of `Managed_design_object` (see 4.2.524). A `Part_terminal_external_reference` is an unambiguous reference to a `Part_terminal` (see 4.2.630) or to a terminal associated with a type of `Part_template` (see 4.2.620) in an external system. A `Part_terminal_external_reference` specifies the type of the terminal as well as the identification for the terminal.

EXAMPLE 1 Items in a design definition may need to reference terminals in a library without duplicating the geometric context, and may need to specify terminal types by reference instead of explicit instantiation.

The data associated with a `Part_terminal_external_reference` are the following:

- `associated_definition`;
- `feature_name`;
- `feature_type`.

4.2.631.1 associated_definition

The `associated_definition` specifies the role of either a `Part_template` (see 4.2.620) or a `Physical_unit` (see 4.2.664) for the `Part_terminal_external_reference`.

4.2.631.2 feature_name

The `feature_name` specifies the role of a string as identification for the `Part_terminal_external_reference`.

4.2.631.3 feature_type

The `feature_type` specifies the string that identifies the type of the terminal referenced for the `Part_terminal_external_reference`. The interpretation of this string by a receiving system shall be in accordance with the interpretation of `Part_terminal`.

Constraints:

UR1: Each member of `Part_terminal_external_reference` shall have a unique combination of attributes.

WR1: The types of terminals are limited to those types defined in this part of ISO 10303.

WR2: If a `Part_terminal` type is specified for the `feature_type`, a `Physical_unit` type shall be specified. If a `Part_terminal` type is not specified for the `feature_type`, a `Part_template` type shall be specified for the `associated_definition`.

NOTE EXPRESS specification:

```

*)
ENTITY part_terminal_external_reference
  SUBTYPE OF (managed_design_object);
    associated_definition : physical_unit_or_part_template;
    feature_name          : ee_name;
    feature_type          : ee_name;
  UNIQUE
    UR1 : associated_definition, feature_name, feature_type;
END_ENTITY; -- part_terminal_external_reference

( *

```

4.2.632 Part_text_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_text_template is a type of Part_template (see 4.2.620) that is the information required to support the creation of text messages in the stratum material on the Interconnect_module (see 4.2.440) or in a marking application.

The data associated with a Part_text_template are the following:

- font;
- maximum_font_horizontal_extent;
- maximum_font_vertical_extent;
- slant_angle;
- stroke_width.

4.2.632.1 font

The font specifies either the Font_definition (see 4.2.387) or the Registered_font (see 4.2.768) for the Part_text_template.

4.2.632.2 maximum_font_horizontal_extent

The maximum_font_horizontal_extent specifies the role of the Length_data_element (see 4.2.511) for the width of the Part_text_template.

4.2.632.3 maximum_font_vertical_extent

The maximum_font_vertical_extent specifies the role of the Length_data_element (see 4.2.511) for the height of the Part_text_template.

4.2.632.4 slant_angle

The slant_angle specifies the Angle_data_element that is the deviation from vertical for the Part_text_template.

4.2.632.5 stroke_width

The stroke_width specifies the Length_data_element that is the distance across a single line that draws the glyph for the Part_text_template.

NOTE A per character shape is required because the message is not available in the template definition. This limits the capability to support proportional font by causing the bounding box to be the maximum size required rather than the minimum that would occur if the bounding box were calculated strictly on the basis of the font proportions in the actual design.

Constraints:

WR1: font glyph shall not extend beyond the shape established by the template_shape.

NOTE EXPRESS specification:

```
*)
ENTITY part_text_template
  SUBTYPE OF (part_template);
    font                : font_select;
    maximum_font_horizontal_extent : length_data_element;
    maximum_font_vertical_extent   : length_data_element;
    slant_angle           : angle_data_element;
    stroke_width          : length_data_element;
END_ENTITY; -- part_text_template

( *
```

4.2.633 Part_tooling_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Part_tooling_feature is a type of Part_feature (see 4.2.609). A Part_tooling_feature may be a Fiducial_part_feature (see 4.2.374) or may be a Test_point_part_feature (see 4.2.894). A Part_tooling_feature identifies a feature for tooling purposes. A feature may be used for other purposes or may intrinsically be in a particular feature category, and in either of those cases the feature would be a Part_tooling_feature and also the other type of feature.

NOTE Tooling purposes as used herein includes test instrumentation and measurement purposes, not just inspection, fabrication and assembly tooling.

EXAMPLE 1 Examples of a Part_tooling_feature include, but are not limited, to slots, passages, notches defined by Geometric Dimensioning and Tolerancing methodology, as well as fiducials.

NOTE EXPRESS specification:

```
*)
ENTITY part_tooling_feature
    SUPERTYPE OF (fiducial_part_feature ANDOR
                  test_point_part_feature)
    SUBTYPE OF (part_feature);
END_ENTITY; -- part_tooling_feature

( *
```

4.2.634 Partially_plated_cutout

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Partially_plated_cutout is a type of Cutout (see 4.2.208) that is not a completely plated cutout for reason of manufacturing capabilities or design purposes.

The data associated with a Partially_plated_cutout are the following:

- plated_segment;
- unplated_segment.

4.2.634.1 plated_segment

A plated_segment is an inverse relationship that specifies that the existence of the Partially_plated_cutout is dependent on the existence of the Plated_cutout_edge_segment that specifies the Partially_plated_cutout as a composed_cutout. There shall be one or more Plated_cutout_edge_segment (see 4.2.686) in the role of plated_segment for the Partially_plated_cutout. The plated_segment specifies the segment that is plated.

4.2.634.2 unplated_segment

An unplated_segment is an inverse relationship that specifies that the existence of the Partially_plated_cutout is dependent on the existence of the Cutout_edge_segment that specifies the Partially_plated_cutout as a composed_cutout. There shall be one or more Cutout_edge_segment (see 4.2.209) in the role of unplated_segment for the Partially_plated_cutout. The unplated_segment specifies the segment that is unplated.

Constraints:

WR1: A Plated_cutout_segment shall not be in the role of unplated_segment.

NOTE EXPRESS specification:

```
*)
ENTITY partially_plated_cutout
  SUBTYPE OF (cutout);
  INVERSE
    plated_segment : SET [1:?] OF cutout_edge_segment FOR composed_cutout;
    unplated_segment : SET [1:?] OF cutout_edge_segment FOR composed_cutout;
END_ENTITY; -- partially_plated_cutout
```

(*

4.2.635 Partially_plated_interconnect_module_edge

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Partially_plated_interconnect_module_edge is a type of Interconnect_module_edge (see 4.2.449) that is not a completely plated edge for reason of manufacturing capabilities or design purposes.

The data associated with a Partially_plated_interconnect_module_edge are the following:

- plated_segment;
- unplated_segment.

4.2.635.1 plated_segment

A plated_segment is an inverse relationship that specifies that the existence of the Partially_plated_interconnect_module is dependent upon the existence of the Plated_interconnect_module_edge_segment that specifies the Partially_plated_interconnect_module_edge in the role of composed_edge. There shall be one or more Plated_interconnect_module_edge_segment (see 4.2.689) for the Partially_plated_interconnect_module_edge.

4.2.635.2 unplated_segment

An unplated_segment is an inverse relationship that specifies that the existence of the Partially_plated_interconnect_module is dependent upon the existence of the Interconnect_module_edge_segment that specifies the Partially_plated_interconnect_module_edge in the role of composed_edge. There shall be one or more Interconnect_module_edge_segment (see 4.2.450) for the Partially_plated_interconnect_module_edge.

Constraints:

WR1: There shall be no Plated_interconnect_module_edge_segment in the role of unplated_segment for the Partially_plated_interconnect_module.

NOTE EXPRESS specification:

```

*)
ENTITY partially_plated_interconnect_module_edge
  SUBTYPE OF (interconnect_module_edge);
INVERSE
  plated_segment : SET [1:?] OF interconnect_module_edge_segment FOR composed_edge;
  unplated_segment : SET [1:?] OF interconnect_module_edge_segment FOR composed_edge;
END_ENTITY; -- partially_plated_interconnect_module_edge

( *
```

4.2.636 Partially_routed_physical_network

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Partially_routed_physical_network is a type of Physical_network (see 4.2.660). A Partially_routed_physical_network is a description of an incompletely specified mapping from requirements to layout. The set of unrouted elements shall be a subset of the total requirement set.

The data associated with a Partially_routed_physical_network are the following:

— unrouted_connectivity_requirement_element.

4.2.636.1 unrouted_connectivity_requirement_element

The unrouted_connectivity_requirement_element specifies the Component_terminal (see 4.2.137) that are not yet implemented by circuitry layout for the Partially_routed_physical_network. There shall be two or more Component_terminal for a Partially_routed_physical_network.

NOTE EXPRESS specification:

```

*)
ENTITY partially_routed_physical_network
  SUBTYPE OF (physical_network);
    unrouted_connectivity_requirement_element : SET [1:?] OF unrouted_join_relationship;
  ( *
WHERE
  WR1: SIZEOF(unrouted_connectivity_requirement_element *
    SELF\physical_network.connectivity_requirement_element) =
    SIZEOF(unrouted_connectivity_requirement_element);
check for true subset of connections
*)
END_ENTITY; -- partially_routed_physical_network
( *

```

4.2.637 Participant_date_and_time_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Participant_date_and_time_assignment is a type of Managed_design_object (see 4.2.524). A Participant_date_and_time_assignment is the relationship between a Person_and_organization (see 4.2.647) and a Date_and_time (see 4.2.213).

The data associated with a Participant_date_and_time_assignment are the following:

- action_date_and_time;
- action_participant.

4.2.637.1 action_date_and_time

The action_date_and_time specifies the role of the Date_and_time (see 4.2.213) for the Participant_date_and_time_assignment.

4.2.637.2 action_participant

The action_participant specifies the role of the Person_and_organization (see 4.2.647) for the Participant_date_and_time_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY participant_date_and_time_assignment
  SUBTYPE OF (managed_design_object);
    action_participant : person_and_organization;
    action_date_and_time : date_and_time;
END_ENTITY; -- participant_date_and_time_assignment

( *
```

4.2.638 Passage_technology

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Passage_technology is a type of Managed_design_object (see 4.2.524). A Passage_technology may be either a Default_component_termination_passage_definition (see 4.2.249), a Default_unsupported_passage_definition (see 4.2.254), or a Default_via_definition (see 4.2.256). A Passage_technology specifies material properties and parametric data associated with a specific manufacturing process for identified material, where that material and manufacturing process has been selected for a specific design. A Passage_technology includes the material stackup parametric data related to the design of the interconnect substrate associated with the Stratum the Passage_technology is related to through the as_finished_inter_stratum_extent attribute.

The data associated with a `Passage_technology` are the following:

- `associated_definition`;
- `as_finished_inter_stratum_extent`;
- `deposition_material`;
- `maximum_aspect_ratio`;
- `maximum_as_finished_deposition_thickness`;
- `maximum_as_finished_passage_extent`;
- `minimum_as_finished_deposition_thickness`;
- `minimum_as_finished_passage_extent`;
- `name`;
- `plated_passage`;
- `specification`.

NOTE `Passage_technology` also supports substrate edges.

4.2.638.1 `associated_definition`

The `associated_definition` specifies the role of the `Interconnect_module` (see 4.2.440) that uses the `Passage_technology`.

NOTE The `Passage_technology` is for a particular design since the vertical extent is within the context of the design.

4.2.638.2 `as_finished_inter_stratum_extent`

The `as_finished_inter_stratum_extent` specifies the `Inter_stratum_extent` (see 4.2.431) that will be created for the `Passage_technology`. Material additions by plating or other means will extend the reach of the `Inter_stratum_extent` to the as-finished condition because these `Stratum` (see 4.2.845) are not present during the machining operation but are created during the finishing operations.

4.2.638.3 `deposition_material`

The `deposition_material` specifies the role of the `Ee_material` (see 4.2.318) that forms the vertical wall of a feature that uses the `Passage_technology`. The `deposition_material` need not be specified for a particular `Passage_technology`.

4.2.638.4 maximum_aspect_ratio

The `maximum_aspect_ratio` specifies the largest allowable ratio of passage length divided by passage diameter for the `Passage_technology`. The `maximum_aspect_ratio` need not be specified for a particular `Passage_technology`.

NOTE The aspect ratio is a factor in determining the expected quality of the electroplating process in depositing metal in the passage. If the aspect ratio is large, meaning an extremely long passage in relation to its diameter, the deposition will be thinner in the central portion of the passage length than at each end of the passage. For this reason, a `maximum_aspect_ratio` is often determined as a factor in deciding upon the smallest passage diameters to be allowed in a PCB design. Thicker PCBs require larger minimum passage diameters, for the same expected plating quality.

4.2.638.5 maximum_as_finished_deposition_thickness

The `maximum_as_finished_deposition_thickness` specifies the `Length_data_element` (see 4.2.511) that is the upper bound of the thickness of the deposition metal in the passage by the `Passage_technology`.

NOTE Material deposition within a passage, by plating or other means, decreases the diameter of the passage. This attribute assists in providing the information required to select the correct tool for creating the passage before plating. The maximum deposition thickness results in the smallest passage diameter after finishing.

4.2.638.6 maximum_as_finished_passage_extent

The `maximum_as_finished_passage_extent` specifies the `Length_data_element` (see 4.2.511) of the upper bound of the diametrical extent after processing for the `Passage_technology`. A `maximum_as_finished_passage_extent` need not be specified for a particular `Passage_technology`.

4.2.638.7 minimum_as_finished_deposition_thickness

The `minimum_as_finished_deposition_thickness` specifies the `Length_data_element` (see 4.2.511) of the lower bound of the thickness of deposited plating in the passage of a PCB for the `Passage_technology`. The `minimum_as_finished_deposition_thickness` need not be specified for a particular `Passage_technology`.

NOTE In order to help insure continuity of the deposited conductive material in a passage, and therefore to help insure connectivity between the plating in a passage and the lands of every `Design_layer_stratum` (see 4.2.275) that intersect that passage, a minimum plating thickness is specified. This plating thickness is usually verified in the fabrication process by including test coupons in the panel of PCBs being fabricated. These coupons include typically sized lands and passages as are found on the PCBs being fabricated; the coupons are removed from the panel during fabrication and metallurgical cross sections are created for process quality verification. These cross sections provide visibility of both the electroless and electrolytic plating process quality as well as the quality of the final tin-lead plating or coating of the passage walls as well as of the so-called knee area, where the passage wall intersects the lands at each end of the passage. Solderability of the tin-lead is partially a factor of the minimum plating thickness at the knee, since tin-lead solder and copper form a visible intermetallic structure when one dissolves into the other during the process of melting the solder. This visible intermetallic structure is brittle and is not indicative of a solder joint of high quality. There must be ample solder, in the bare PCB, over this intermetallic knee, in order to assist in providing a PCB that has good solderability to the assembly operation.

4.2.638.8 minimum_as_finished_passage_extent

The `minimum_as_finished_passage_extent` specifies the `Length_data_element` (see 4.2.511) of the lower bound of the diametrical extent for the `Passage_technology`. The `minimum_as_finished_passage_extent` need not be specified for a particular `Passage_technology`.

4.2.638.9 name

The name identifies the `Passage_technology`.

4.2.638.10 plated_passage

The `plated_passage` specifies whether or not the `deposition_material` is added to the passage for the `Passage_technology`. The value of TRUE indicates that the material is added. The value of FALSE indicates that the material is not added.

NOTE 1 This boolean makes no reference to the process to be used for adding a tin-lead alloy or other solder to the conductive walls of the `plated_passage`. Some processes electroplate the tin-lead alloy and later melt, or reflow, the alloy to reduce its granular appearance; other processes coat the passage walls and other exposed conductive surfaces of the PCB with molten solder in a process called solder-coating. These processes are only applicable to `plated_passages` in the PCB.

NOTE 2 This part of ISO 10303 does not presume a `Pcb` (see 4.2.642) wet-chemistry only process, but allows for other processes of metallization of passage walls.

4.2.638.11 specification

The specification establishes the role of `Process_specification` for the `Passage_technology`. The specification need not be specified for a particular `Passage_technology`.

Constraints:

UR1: The name shall be unique within `associated_definition`.

WR1: Each passage that is plated shall have `deposition_material` provided.

WR2: Each passage that has `maximum_as_finished_deposition_thickness` provided shall have `deposition_material` provided.

WR3: Each passage that has `minimum_as_finished_deposition_thickness` provided shall have `deposition_material` provided.

WR4: If `plated_passage` has a value of TRUE `Passage_technology` shall reference `Material_conductivity_classification` that equals "conductive".

NOTE EXPRESS specification:

```

*)
ENTITY passage_technology
  SUPERTYPE OF (ONEOF (default_via_definition,
    default_component_termination_passage_definition,
    default_unsupported_passage_definition))
  SUBTYPE OF (managed_design_object);
  associated_definition                : interconnect_module;
  as_finished_inter_stratum_extent    : inter_stratum_extent;
  deposition_material                  : OPTIONAL ee_material;
  plated_passage                       : BOOLEAN;
  maximum_as_finished_passage_extent  : OPTIONAL length_data_element;
  minimum_as_finished_passage_extent  : OPTIONAL length_data_element;
  name                                 : ee_name;
  maximum_as_finished_deposition_thickness : OPTIONAL length_data_element;
  maximum_aspect_ratio                 : OPTIONAL REAL;
  minimum_as_finished_deposition_thickness : OPTIONAL length_data_element;
  specification                         : OPTIONAL process_specification;
  UNIQUE
    UR1 : name, associated_definition;
END_ENTITY; -- passage_technology
(*)

```

4.2.639 Pca

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pca is a type of Assembly_module (see 4.2.43). A Pca describes a printed circuit assembly. A printed circuit assembly is nearly a synonym for printed wiring assembly, except that a printed circuit assembly may include printed components in its printed circuit board and a printed wiring assembly is constructed upon a printed wiring board that is totally interconnect without printed components. Another synonym for printed circuit assembly is circuit card assembly.

EXAMPLE 1 See figure 75 for an illustration of a Pca.

The data associated with a Pca are the following:

— usage_view.

4.2.639.1 usage_view

The usage_view specifies the Pca_usage_view (see 4.2.641), as seen from the context of its next assembly, for the Pca.

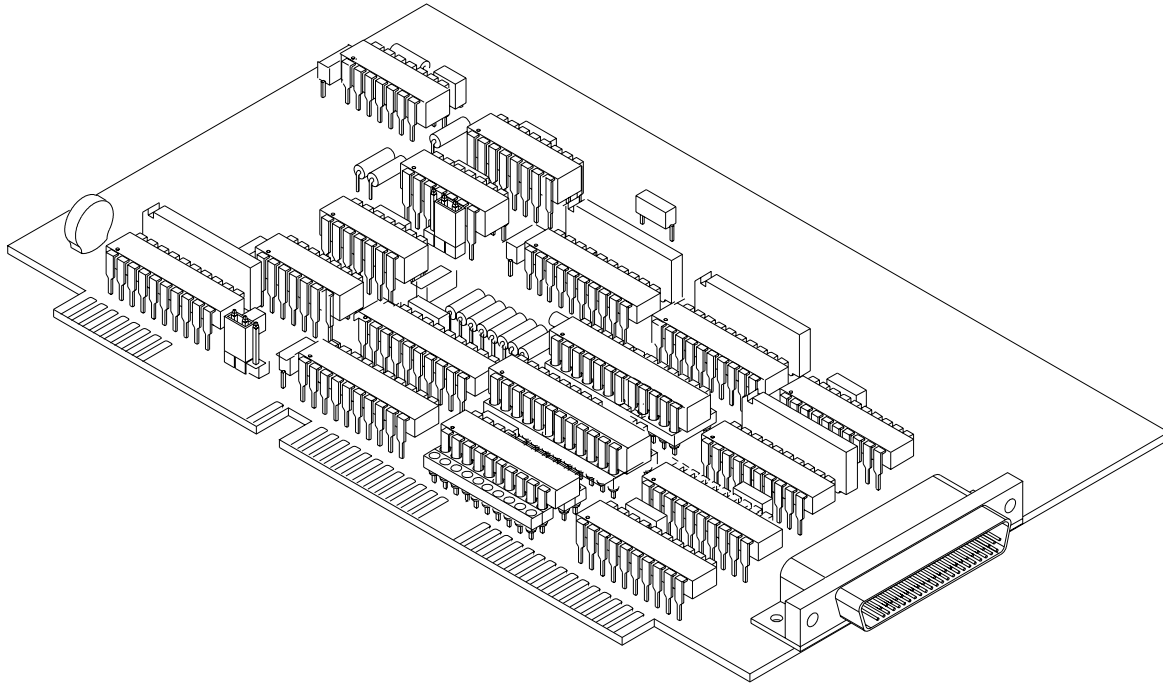


Figure 75 – Pca

NOTE EXPRESS specification:

```
* )
ENTITY pca
  SUBTYPE OF (assembly_module);
  SELF\physical_unit_design_view.usage_view : pca_usage_view;
END_ENTITY; -- pca
```

(*

4.2.640 Pca_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pca_terminal is a type of Assembly_module_terminal (see 4.2.50) that is part of the interface between the Pca (see 4.2.639) and its next higher assembly.

The data associated with a Pca_terminal are the following:

— associated_definition.

4.2.640.1 associated_definition

The associated_definition specifies the role of the Pca_usage_view (see 4.2.641) for the Pca_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY pca_terminal
  SUBTYPE OF (assembly_module_terminal);
  SELF\part_feature.associated_definition : pca_usage_view;
END_ENTITY; -- pca_terminal
```

(*

4.2.641 Pca_usage_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pca_usage_view is a type of Assembly_module_usage_view (see 4.2.51). A Pca_usage_view provides the information needed to include the Pca (see 4.2.639) in the design of the next higher level assembly, but not necessarily sufficient information to reproduce that Pca.

NOTE A Packaged_connector may be located in a usage view in addition to terminals. Reference Assembly_module_usage_view_connector_relationship (see 4.2.52)

EXAMPLE 1 Figure 76 illustrates details available in a usage view for a Pca. The information available in the usage view specifically is separate from that in the design view and the relationships between the two views are usually controlled by a design organization.

Constraints:

WR1: Each Pca_usage_view shall have at least two terminals.

NOTE EXPRESS specification:

```
*)
ENTITY pca_usage_view
  SUBTYPE OF (assembly_module_usage_view);
END_ENTITY; -- pca_usage_view
```

(*

4.2.642 Pcb

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

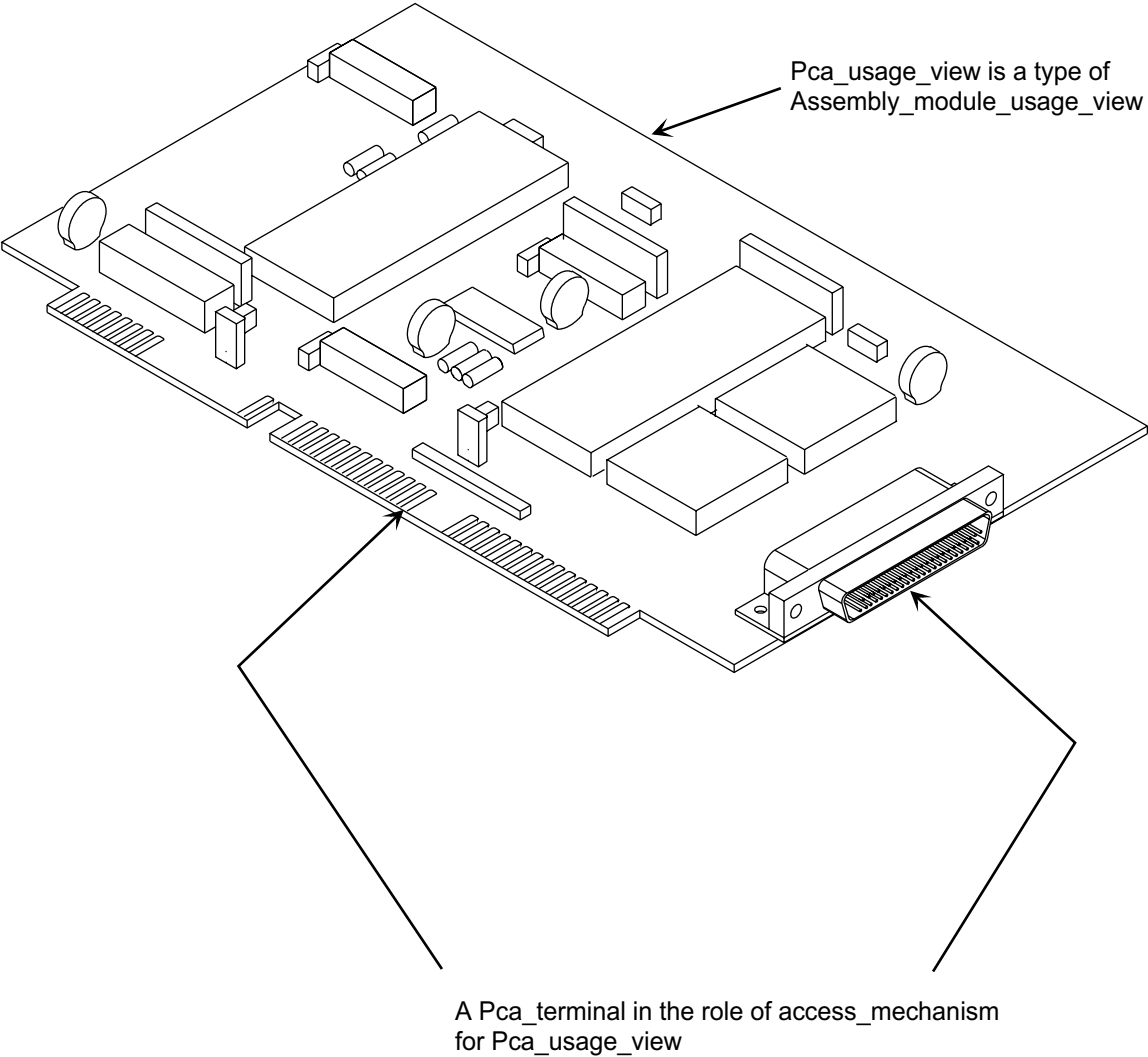
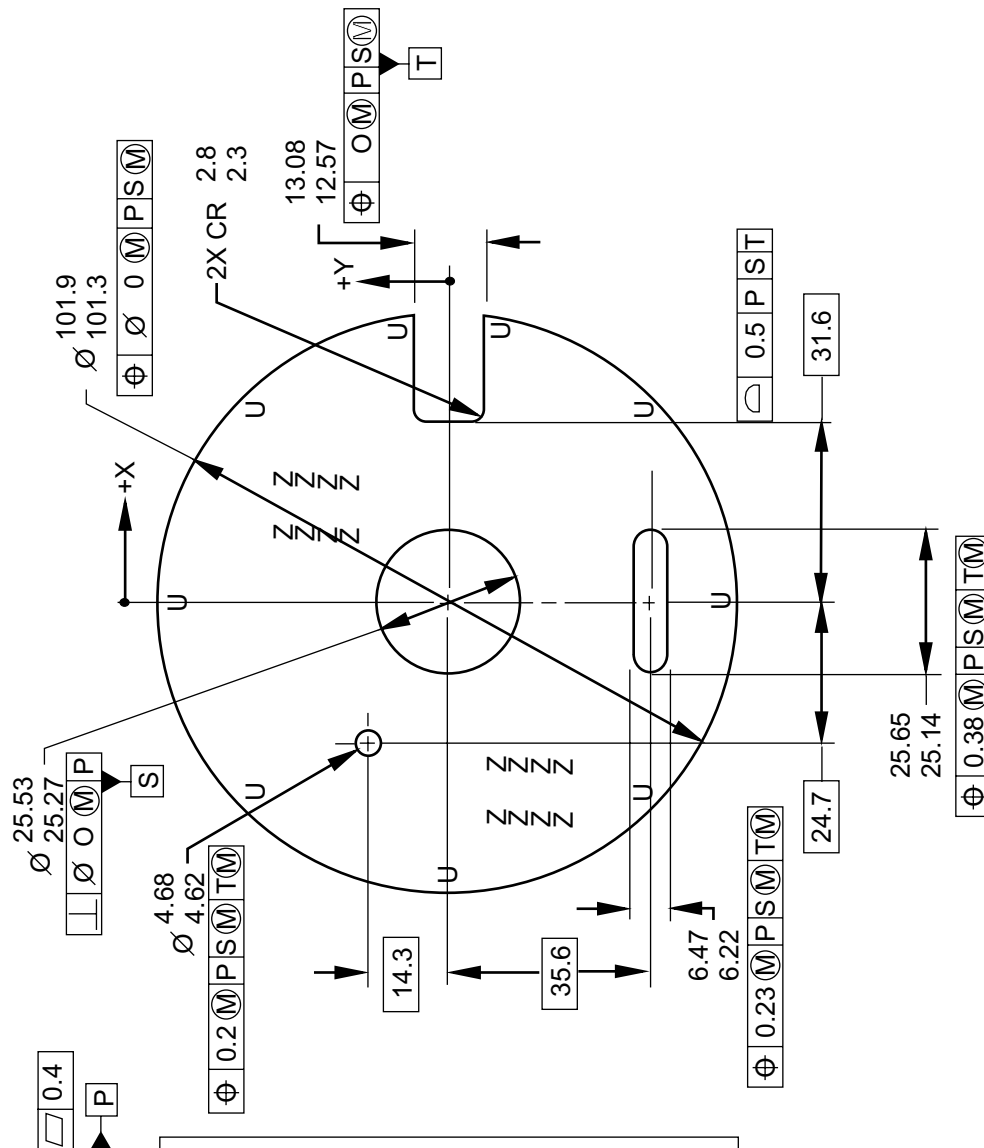


Figure 76 – Pca_usage_view

A Pcb is a type of Interconnect_module (see 4.2.440). A Pcb describes a printed circuit board. A printed circuit board is nearly a synonym for printed wiring board, except that a printed circuit board may include printed components and a printed wiring board is totally interconnected without printed components. This part of ISO 10303 supports the exchange of product categories in order to allow the exchange partners to differentiate between printed wiring board and printed circuit board.

EXAMPLE 1 Figure 79 illustrates a printed wiring board design.

EXAMPLE 2 Figures 77 and 78 illustrate complex printed wiring board mechanical designs that have functional shape requirements that are communicated using Geometric Dimensioning and Tolerancing information in a product model and not graphically.



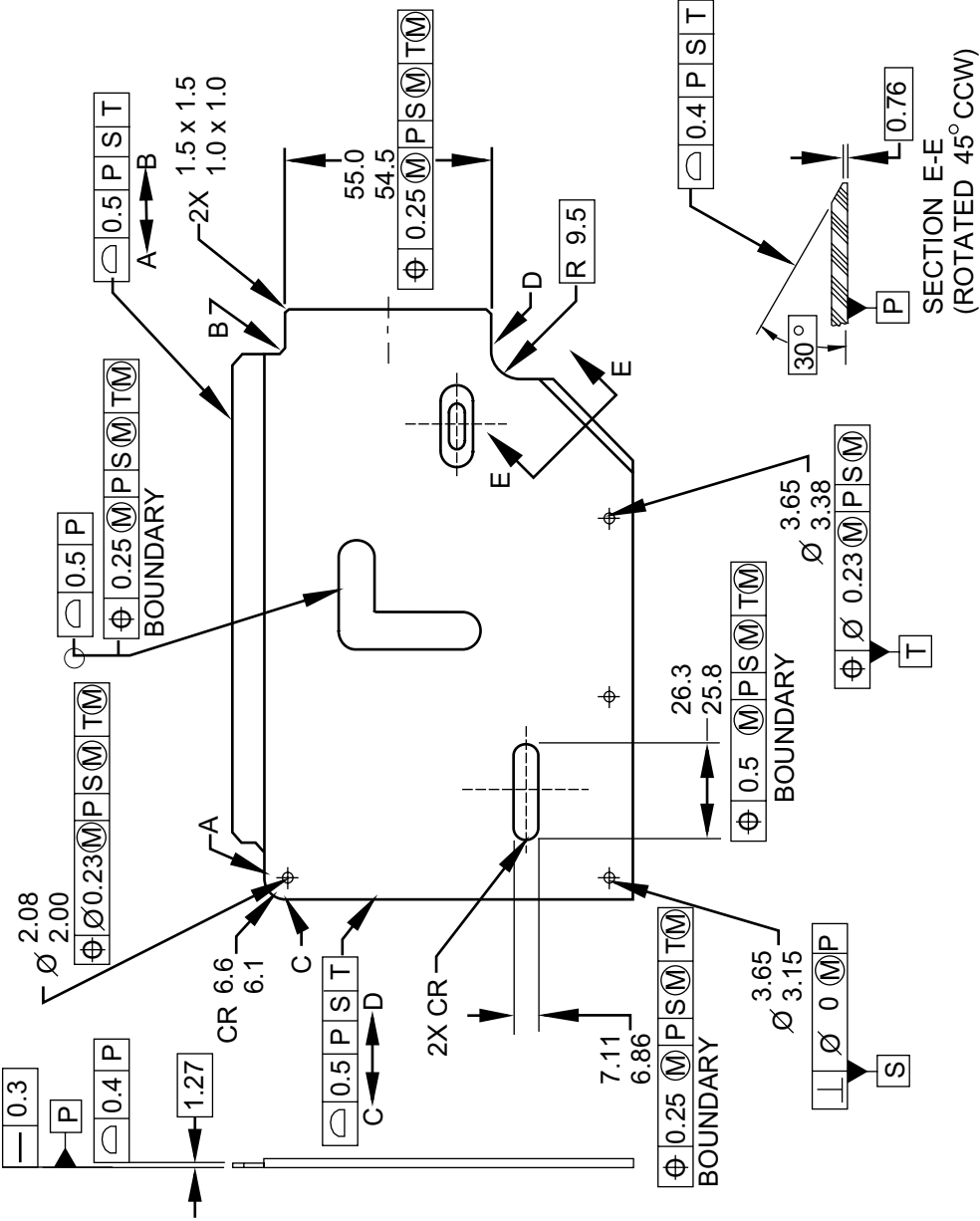


Figure 78 – Complex Pcb tolerances

The data associated with a Pcb are the following:

— usage_view.

4.2.642.1 usage_view

The usage_view specifies the role of the Pcb_usage_view (see 4.2.643) for the Pcb. The usage_view shall not be specified if the Pcb is a Layout_macro_definition, otherwise it shall exist.

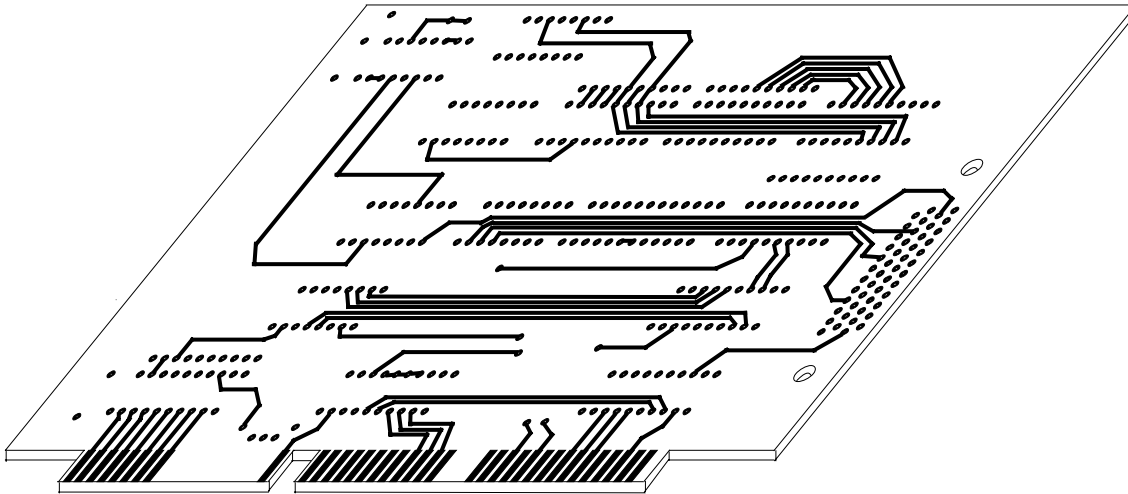


Figure 79 – Bare printed wiring board

NOTE EXPRESS specification:

```

*)
ENTITY pcb
  SUPERTYPE OF (layout_macro_definition)
  SUBTYPE OF (interconnect_module);
  SELF\physical_unit_design_view.usage_view : OPTIONAL pcb_usage_view;
WHERE
  WR1: ('AP210_ARM.LAYOUT_MACRO_DEFINITION' IN TYPEOF(SELF)) XOR
    (EXISTS(SELF\physical_unit_design_view.usage_view));
END_ENTITY; -- pcb

```

(*

4.2.643 Pcb_usage_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Pcb_usage_view` is a type of `Interconnect_module_usage_view` (see 4.2.465) that provides the information necessary to include the `Pcb` (see 4.2.642) in its next higher level assembly for design and some simulation purposes, but may not include sufficient information to reproduce the `Pcb`.

EXAMPLE 1 Some of the information in figures 77 and 78 that communicate fit tolerances may be included in a usage view.

NOTE EXPRESS specification:

```
*)
ENTITY pcb_usage_view
  SUBTYPE OF (interconnect_module_usage_view);
END_ENTITY; -- pcb_usage_view

( *
```

4.2.644 Perpendicular_shape_element_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Perpendicular_shape_element_relationship` is a type of `Shape_element_relationship` (see 4.2.816) that asserts that two `Shape_element` (see 4.2.810) are perpendicular to each other. One `Shape_element` is specified as the `related_shape_element` and the other `Shape_element` is specified as the `relating_shape_element`.

NOTE 1 The attribute `related_shape_element` is declared in the `Shape_element_relationship` entity of which this entity is a subtype.

NOTE 2 The attribute `relating_shape_element` is declared in the `Shape_element_relationship` entity of which this entity is a subtype.

Constraints:

WR1: The `Shape_element` specified as the `relating_shape_element` shall not be specified as perpendicular to itself either directly or through intermediate `Perpendicular_shape_element_relationship`.

NOTE EXPRESS specification:

```
*)
ENTITY perpendicular_shape_element_relationship
  SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- perpendicular_shape_element_relationship

( *
```

4.2.645 Perpendicularity_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Perpendicularity_tolerance is a type of Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that specifies or partially specifies the design requirement on the perpendicularity of a Shape_element (see 4.2.810) in relation to one or two Datum (see 4.2.214).

NOTE Perpendicularity tolerance is defined in terms of tolerance zone definitions in 14.8 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Centre_axis, Centre_plane, Physical_feature_or_part_template, or Tangent_plane.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: There shall be at most two Datum in the Datum_system the definition of which is specified as the datum_system_definition.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

WR3: At least one Datum_plane or Datum_axis shall be in the Datum_system the definition of which is specified as the datum_system_definition.

WR4: If the toleranced_shape_element is a Centre_plane, Physical_feature_or_part_template, or Tangent_plane, the Perpendicularity_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

IP1: The form of the toleranced Shape_element shall be consistent with the form of a shape element for which the application of a perpendicularity tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- perpendicularity_tolerance
```

```
( *
```

4.2.646 Person

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Person is a type of Managed_design_object (see 4.2.524). A Person is an individual within the context of a single Organization (see 4.2.569).

The data associated with a Person are the following:

- address;
- identifier.

4.2.646.1 address

The address specifies the string for the interpretable name for the location of the Person. The address specifies the routing for postal mail or the physical location of the Person. The address need not be specified for a particular Person.

4.2.646.2 identifier

The identifier specifies the name that represents either the personal name or title of the Person.

NOTE EXPRESS specification:

```
*)
ENTITY person
  SUBTYPE OF (managed_design_object);
    identifier : ee_name;
    address    : OPTIONAL STRING;
END_ENTITY; -- person

( *
```

4.2.647 Person_and_organization

A Person_and_organization is a type of Managed_design_object (see 4.2.524). A Person_and_organization is the aggregation of a Person (see 4.2.646) and an Organization (see 4.2.569).

The data associated with a Person_and_organization are the following:

- assigned_organization;
- assigned_person;
- person_id_in_organization.

4.2.647.1 assigned_organization

The assigned_organization specifies the role of the Organization (see 4.2.569) for the Person_and_organization.

4.2.647.2 assigned_person

The assigned_person specifies the role of the Person (see 4.2.646) for the Person_and_organization.

4.2.647.3 person_id_in_organization

The person_id_in_organization specifies the identifier for the Person within the context of the Organization for the Person_and_organization. The person_in_organization is optional and need not be supplied.

Constraints:

UR1: The combination of the person_id_in_organization and the assigned_organization shall be unique within a population of Person_and_organizations.

UR2: The combination of the assign_person and the assigned_organization shall be unique within a population of Person_and_organizations.

NOTE EXPRESS specification:

```
* )
ENTITY person_and_organization
  SUBTYPE OF (managed_design_object);
    assigned_person      : person;
    assigned_organization : organization;
    person_id_in_organization : OPTIONAL ee_name;
  UNIQUE
    UR1 : assigned_person, assigned_organization;
    UR2 : person_id_in_organization, assigned_organization;
END_ENTITY; -- person_and_organization
(*
```

4.2.648 Physical_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_component is a type of Assembly_component (see 4.2.34). A Physical_component is one of a Cable_component (see 4.2.84), a Packaged_component (see 4.2.585), an Interconnect_module_component (see 4.2.442), an Assembly_module_component (see 4.2.44), or a Bare_die_component (see 4.2.61), or additionally may be an Interface_component (see 4.2.468).

NOTE EXPRESS specification:

```

*)
ENTITY physical_component
  SUPERTYPE OF (ONEOF (
--tt tc1 wd12 modified
    cable_component,
    packaged_component,
    interconnect_module_component,
    assembly_module_component,
    bare_die_component,
--tt tc1 wd12 modified
    routed_physical_component,
    routed_interconnect_component)
  ANDOR interface_component)
  SUBTYPE OF (assembly_component);
  SELF\assembly_component.usage_definition : OPTIONAL physical_unit;
END_ENTITY; -- physical_component

( *
```

4.2.649 Physical_component_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_component_feature` is a type of `Component_feature` (see 4.2.118) and a type of `Physical_feature_or_part_template` (see 4.2.658). The `Physical_component_feature` supports the specification of a `Component_feature` as a `Datum_target` or `Datum_feature` for the purpose of establishing a `Datum_reference_frame` for tolerance specification.

NOTE EXPRESS specification:

```

*)
ENTITY physical_component_feature
--tt tc1 wd8 modified
--tt tc1 wd12 modified
  ABSTRACT SUPERTYPE OF (component_mounting_feature ANDOR
    interconnect_module_component_surface_feature ANDOR
    (ONEOF(physical_component_terminal, component_interface_terminal)))
  SUBTYPE OF (component_feature, physical_feature_or_part_template);
END_ENTITY; -- physical_component_feature

--tt tc1 wd12 added
( *
```

4.2.650 Physical_component_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_component_terminal` is a

The data associated with a `Physical_component_terminal` are the following:

— `additional_data`.

4.2.650.1 `additional_data`

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY physical_component_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (
    assembly_module_component_terminal,
    bare_die_component_terminal,
    interconnect_component_join_terminal,
    interconnect_module_component_terminal,
    minimally_defined_component_terminal,
    packaged_component_join_terminal))
  SUBTYPE OF (component_terminal, physical_component_feature);
INVERSE
  connected_node      : SET [0:1] OF physical_connectivity_definition FOR
                        associated_terminals;
END_ENTITY; -- physical_component_terminal

( *
```

4.2.651 `Physical_connectivity_abstraction_map`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_connectivity_abstraction_map` is a type of `Managed_design_object` (see 4.2.524) that provides traceability between a `Physical_connectivity_element` (see 4.2.654) and a `Conductive_interconnect_element` (see 4.2.171) for simulation or routing purposes.

The data associated with a `Physical_connectivity_abstraction_map` are the following:

— `geometric_context_based_definition`;

— `name_space_context_based_definition`.

4.2.651.1 geometric_context_based_definition

The `geometric_context_based_definition` specifies the role of the `Conductive_interconnect_element` (see 4.2.171) for the `Physical_connectivity_abstraction_map`.

4.2.651.2 name_space_context_based_definition

The `name_space_context_based_definition` specifies the role of the `Physical_connectivity_element` (see 4.2.654) for the `Physical_connectivity_abstraction_map`.

Constraints:

UR1: The combination of the `geometric_context_based_definition` and the `name_space_context_based_definition` shall be unique within a population of `Physical_connectivity_abstraction_maps`.

NOTE EXPRESS specification:

```

*)
ENTITY physical_connectivity_abstraction_map
  SUBTYPE OF (managed_design_object);
    name_space_context_based_definition : physical_connectivity_element;
    geometric_context_based_definition  : conductive_interconnect_element;
UNIQUE
  UR1: name_space_context_based_definition, geometric_context_based_definition;
END_ENTITY; -- physical_connectivity_abstraction_map

( *
```

4.2.652 Physical_connectivity_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_connectivity_definition` is a type of `Managed_design_object` (see 4.2.524). A `Physical_connectivity_definition` may be a `Physical_connectivity_structure_definition` (see 4.2.657). A `Physical_connectivity_definition` is the root node in the classification hierarchy of connectivity in the physical domain. All types of connectivity in this domain contain additional information to that of the base type. A `Physical_connectivity_definition` is a definition of one node, in an electrical network, consisting of physical terminals.

The data associated with a `Physical_connectivity_definition` are the following:

- `associated_terminals`;
- `name`.

4.2.652.1 associated_terminals

The associated_terminals specify the Component_terminal (see 4.2.137) of the Physical_connectivity_definition. There shall be two or more Component_terminal for a Physical_connectivity_definition.

4.2.652.2 name

The name identifies the Physical_connectivity_definition.

Constraints:

UR1: The name shall be unique within the context of the associated_definition.

WR1: Each design view that references a member of associated_terminals shall reference all members of associated_terminals.

WR2: The associated_definition shall reference all members of associated_terminals.

NOTE EXPRESS specification:

```

*)
ENTITY physical_connectivity_definition
  SUPERTYPE OF (physical_connectivity_structure_definition)
  SUBTYPE OF (managed_design_object);
  associated_definition : physical_unit_network_definition;
  associated_terminals : SET [2:?] OF physical_component_terminal;
  name                  : ee_name;
  UNIQUE
  UR1 : name, associated_definition;
END_ENTITY; -- physical_connectivity_definition
( *
```

4.2.653 Physical_connectivity_definition_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_connectivity_definition_relationship is a type of Managed_design_object (see 4.2.524). A Physical_connectivity_definition_relationship is the association between Physical_connectivity_definition (see 4.2.652) that exists because of a Requirement_occurrence.

NOTE The purpose of the Physical_connectivity_definition_relationship is documented in the Requirement_occurrence, and is typically life cycle dependent, based on design iterations, or variations.

The data associated with a `Physical_connectivity_definition_relationship` are the following:

- `current_definition`;
- `previous_definition`;
- `requirement`.

4.2.653.1 `current_definition`

The `current_definition` specifies one role of the `Physical_connectivity_definition` (see 4.2.652) for the `Physical_connectivity_definition_relationship`.

4.2.653.2 `previous_definition`

The `previous_definition` specifies one role of the `Physical_connectivity_definition` (see 4.2.652) for the `Physical_connectivity_definition_relationship`.

4.2.653.3 `requirement`

The `requirement` specifies the `Ee_requirement_occurrence` (see 4.2.330) that causes the `Physical_connectivity_definition_relationship` to be in existence between the current `Physical_connectivity_definition` (see 4.2.652) and the previous `Physical_connectivity_definition`.

NOTE EXPRESS specification:

```

*)
ENTITY physical_connectivity_definition_relationship
  SUBTYPE OF (managed_design_object);
    requirement      : ee_requirement_occurrence;
    previous_definition : physical_connectivity_definition;
    current_definition : physical_connectivity_definition;
END_ENTITY; -- physical_connectivity_definition_relationship

( *
```

4.2.654 `Physical_connectivity_element`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_connectivity_element` is a type of `Managed_design_object` (see 4.2.524). A `Physical_connectivity_element` is the object that describes a link between two nodes for the graph representation.

The data associated with a `Physical_connectivity_element` are the following:

- `connectivity_context`;
- `end_terminus`;
- `name`;
- `start_terminus`.

4.2.654.1 `connectivity_context`

The `connectivity_context` specifies the role of the `Physical_connectivity_structure_definition` (see 4.2.657) for the `Physical_connectivity_element`. The `connectivity_context` contains all the link elements of the graph, of which a `Physical_connectivity_element` is one.

4.2.654.2 `end_terminus`

The `end_terminus` specifies either a `Component_terminal` (see 4.2.137) or a `Topological_junction` (see 4.2.928) as the end of the `Physical_connectivity_element`.

4.2.654.3 `name`

The `name` identifies the member of `Physical_connectivity_element` within the context of the assembly.

4.2.654.4 `start_terminus`

The `start_terminus` specifies either a `Component_terminal` (see 4.2.137) or a `Topological_junction` (see 4.2.928) as the beginning of the `Physical_connectivity_element`.

Constraints:

UR1: The combination of the `name` and the `connectivity_context` shall be unique within a population of `Physical_connectivity_element`.

WR1: The `start_terminus` shall not be the `end_terminus`;

WR2: Each object referenced by the `Physical_connectivity_element` shall be in the design view specified by `connectivity_context.associated_definition`.

NOTE EXPRESS specification:

```

*)
ENTITY physical_connectivity_element
  SUBTYPE OF (managed_design_object);
    start_terminus      : termination_or_junction;
    name                 : ee_name;
    end_terminus        : termination_or_junction;
    connectivity_context : physical_connectivity_structure_definition;
  UNIQUE
    UR1 : name, connectivity_context;
END_ENTITY; -- physical_connectivity_element

```

(*

4.2.655 Physical_connectivity_interrupting_cutout

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_connectivity_interrupting_cutout is a type of Cutout (see 4.2.208) that is included in the design of an Interconnect_module (see 4.2.440) for the purpose of interrupting one or more connectivity_elements in an Interconnect_module.

NOTE A Physical_connectivity_interrupting_cutout is used where a relayout of the design that would remove the connectivity elements from the design is not provided, but where traceability to a Physical_connectivity_definition (see 4.2.652) is required.

The data associated with a Physical_connectivity_interrupting_cutout are the following:

— interrupted_connectivity_element.

4.2.655.1 interrupted_connectivity_element

The interrupted_connectivity_element specifies the Conductive_interconnect_element (see 4.2.171) that is interrupted with the Physical_connectivity_interrupting_cutout. There shall be one or more Conductive_interconnect_element for a Physical_connectivity_interrupting_cutout.

NOTE EXPRESS specification:

```

*)
ENTITY physical_connectivity_interrupting_cutout
  SUBTYPE OF (cutout);
    interrupted_connectivity_element : SET [1:?] OF conductive_interconnect_element;
END_ENTITY; -- physical_connectivity_interrupting_cutout

```

(*

4.2.656 **Physical_connectivity_make_from_relationship**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_connectivity_make_from_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Physical_connectivity_make_from_relationship` is an association between one `Physical_connectivity_definition` (see 4.2.652) that is used as a basis for modification into another `Physical_connectivity_definition` and the modified `Physical_connectivity_definition` that results from the modification.

The data associated with a `Physical_connectivity_make_from_relationship` are the following:

- `resultant_definition`;
- `reusable_definition`.

4.2.656.1 **resultant_definition**

The `resultant_definition` specifies the `Physical_connectivity_definition` (see 4.2.652) that results from modification of the `reusable_definition` of the `Physical_connectivity_make_from_relationship`.

4.2.656.2 **reusable_definition**

The `reusable_definition` specifies the `Physical_connectivity_definition` (see 4.2.652) for the basis for modification into the `resultant_definition` in the `Physical_connectivity_make_from_relationship`.

NOTE EXPRESS specification:

```
*)
ENTITY physical_connectivity_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_definition : physical_connectivity_definition;
    resultant_definition : physical_connectivity_definition;
END_ENTITY; -- physical_connectivity_make_from_relationship

( *
```

4.2.657 **Physical_connectivity_structure_definition**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_connectivity_structure_definition` is a type of `Physical_connectivity_definition` (see 4.2.652) that has a structure. Structure included supports assignment of path based properties to a `Physical_connectivity_definition` for routing or validation purposes. The structure is a graph and may be a tree if there are no redundant arcs, but that is not a requirement.

The data associated with a `Physical_connectivity_structure_definition` are the following:

— `structural_element`.

4.2.657.1 `structural_element`

The `structural_element` is an inverse relationship that specifies that the existence of the `Physical_connectivity_structure_definition` is dependent on the existence of the `Physical_connectivity_element` (see 4.2.654) that specifies the `Physical_connectivity_structural_definition` as its `connectivity_context`. There shall be one or more `Physical_connectivity_element` for a `Physical_connectivity_structural_definition`.

NOTE Ordering is not explicitly represented in this Application Object.

NOTE EXPRESS specification:

```
*)
ENTITY physical_connectivity_structure_definition
  SUBTYPE OF (physical_connectivity_definition);
  INVERSE
    structural_element : SET [1:?] OF physical_connectivity_element FOR
                          connectivity_context;
END_ENTITY; -- physical_connectivity_structure_definition

( *
```

4.2.658 `Physical_feature_or_part_template`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_feature_or_part_template` is a type of `Shape_element` (see 4.2.810) that is either on the physical boundary of a `Physical_unit` (see 4.2.664) or is a `Part_template` (see 4.2.620). A `Physical_feature_or_part_template` is one of a `Part_feature` (see 4.2.609), a `Physical_component_feature` (see 4.2.649), a `Physical_laminate_component` (see 4.2.659), a `Part_template`, or a `Stratum_feature` (see 4.2.851). Additionally, a `Physical_feature_or_part_template` may either be a `Datum_feature` (see 4.2.226) or a `Datum_target` (see 4.2.240). The `Physical_feature_or_part_template` is an intersection application object that supports assignment of Geometric Dimensioning and Tolerance data to features or template definitions.

Constraints:

IP1: If the `Physical_feature_or_part_template` is not a `Part_template` it shall be on the boundary of the `Physical_unit`.

NOTE EXPRESS specification:

```

*)
ENTITY physical_feature_or_part_template
  ABSTRACT SUPERTYPE OF (ONEOF (part_feature,
    physical_component_feature, physical_laminate_component, part_template,
    stratum_feature) ANDOR ONEOF (datum_feature, datum_target))
  SUBTYPE OF (shape_element);
END_ENTITY; -- physical_feature_or_part_template

( *

```

4.2.659 Physical_laminate_component

A `Physical_laminate_component` is a type of `Laminate_component` (see 4.2.483) and a type of `Physical_feature_or_part_template` (see 4.2.658). The `Physical_laminate_component` supports the specification of a `Laminate_component` as a `Datum_target` or `Datum_feature` for the purpose of establishing a `Datum_reference_frame` for tolerance specification. A `Physical_laminate_component` shall be in a complex instance with another member of the subtype tree of `Laminate_component` and `Physical_feature_or_part_template`. The interpretation shall be in compliance with the interpretation as if `Physical_laminate_component` were `Abstract`.

NOTE ISO 10303-11 does not permit `Physical_laminate_component` to be `Abstract`. Therefore this requirement is encoded in a rule.

NOTE EXPRESS specification:

```

*)
ENTITY physical_laminate_component
  SUBTYPE OF (laminate_component, physical_feature_or_part_template);
WHERE
  WR1: SIZEOF(TYPEOF(SELF)) &#62; 6;
END_ENTITY; -- physical_laminate_component

( *

```

4.2.660 Physical_network

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_network` is a type of `Managed_design_object` (see 4.2.524). A `Physical_network` may be a `Partially_routed_physical_network` (see 4.2.636). A `Physical_network` is the result of mapping a topological definition of a physical connectivity requirement onto the conducting `Design_layer_stratum` (see 4.2.275) of an `Interconnect_module` (see 4.2.440). The vertices, specified by `Component_terminal` (see 4.2.137), are mapped onto `Layer_connection_point` (see 4.2.499) on the surface of one or more conducting `Design_layer_stratum` in the `Interconnect_module`. The design features that are implementing the `Physical_network` are accessed by traversing the structure specified by the `interconnect_module_net_topology_element` attribute. The `Physical_network` shall not implement direct connections to `Component_terminal` specified by `Interface_mounted_join` (see 4.2.470), or `Assembly_joint` (see 4.2.41), but shall implement connections only to `Component_terminals` in the scope of the design view being implemented. This `Physical_network` implements connectivity to the join terminals of the components that are interface components in this `Interconnect_module`, which is a design view. The scope of the `Physical_network` is entirely within the interconnect design view and provides no information about assembly connectivity of design views within which occurrences of the `Interconnect_module`, as `Interconnect_module_component` (see 4.2.442) may appear. The `Physical_network` shall not use as `connectivity_requirement_element` terminals of `Land` or `Non_functional_land` that are supporting internal implementation of the `Physical_network`. Only `Land` or `Non_functional_land` types that have `interface_terminals` shall be referenced.

NOTE The `Physical_node_requirement_to_implementing_component_allocation` (see 4.2.663) application object is provided so that traceability of the required connectivity for an assembly design view to that implemented internally in an `Interconnect_module` may be supported. The `Part_connected_terminals_definition` (see 4.2.605) application object is provided to specify explicitly the external terminals of the interconnect substrate that are to be connected. In a case where a design organization re-uses an interconnect design, the `Part_connected_terminals_definition`, since it is the actual connection statement for an `Interconnect_module`, defines the capabilities of the interconnect substrate to satisfy the requirements imposed by the `Physical_node_requirement_to_implementing_component_allocation` object. There may be many sets of `Physical_node_requirement_to_implementing_component_allocation` (one set per assembly design) that reference an interconnect substrate. There is only one set of `Part_connected_terminals_definition` for an interconnect substrate.

The data associated with a `Physical_network` are the following:

- `current_associated_definition`;
- `prior_associated_definition`;
- `connectivity_requirement_element`;
- `interconnect_module_net_topology_element`;
- `name`.

4.2.660.1 `current_associated_definition`

The `current_associated_definition` specifies the `Interconnect_module` (see 4.2.440) that is the current product definition associated with the `Physical_network`.

4.2.660.2 prior_associated_definition

The prior_associated_definition specifies one or more Interconnect_module (see 4.2.440) that are definitions for prior versions of the design. The prior_associated_definition need not be provided.

4.2.660.3 connectivity_requirement_element

The connectivity_requirement_element specifies the Component_terminal (see 4.2.137) to be connected by the network described by the Physical_network. There shall be two or more Component_terminal for a Physical_network.

4.2.660.4 interconnect_module_net_topology_element

The interconnect_module_net_topology_element specifies the unique Join_relationship (see 4.2.479) that results from mapping the topological statement of the physical connectivity requirement onto the coordinate space of the conducting Design_layer_stratum (see 4.2.275) that compose the Interconnect_module (see 4.2.440). There shall be one or more Join_relationship for a Physical_network.

4.2.660.5 name

The name identifies the Physical_network.

Constraints:

UR1: The name shall be unique within a population of Physical_networks in the context of the current_associated_definition.

WR1: The Component_terminal specified by connectivity_requirement_element shall be one of the following:

Land_join_terminal,

Non_functional_land_join_terminal,

Printed_component_join_terminal,

Via_terminal,

Component_termination_passage_join_terminal.

WR2: All members of connectivity_requirement_element and interconnect_module_network_topology_element shall be in the current_associated_definition.

WR3: If the prior_associated_definition exists, then all members of connectivity_requirement_element and interconnect_module_network_topology_element shall be in the prior_associated_definition.

WR4: If the prior_associated_definition exists, then each member of prior_associated_definition shall not be an current_associated_definition.

WR5: Each Ee_material that participates in the Physical_network shall be the same material.

WR6: Each member of Component_terminal in the role of connectivity_requirement_element and each member of Join_relationship in the role of interconnect_module_network_topology_element referenced by Physical_network shall be in the same design view or shall be an explicit external reference.

NOTE A Physical_network does not include connection requirements that are implemented by a Printed_component.

NOTE EXPRESS specification:

```

*)
ENTITY physical_network
  SUPERTYPE OF (partially_routed_physical_network)
  SUBTYPE OF (managed_design_object);
    current_associated_definition          : interconnect_module;
    prior_associated_definition            : OPTIONAL SET [1:?] OF interconnect_modu
    connectivity_requirement_element        : SET [2:?] OF laminate_component_join_te
    interconnect_module_network_topology_element : SET [1:?] OF join_relationship;
    name                                   : ee_name;
    reference_connected_terminals           : OPTIONAL part_connected_terminals_defin
  UNIQUE
    UR1 : name, current_associated_definition;
WHERE
  WR1: NOT(EXISTS(prior_associated_definition)) OR
    NOT(current_associated_definition IN prior_associated_definition);
  WR2: SIZEOF(QUERY(cre &#60;* connectivity_requirement_element |
    NOT(NOT('AP210_ARM.LAND_JOIN_TERMINAL' IN TYPEOF(cre))
      OR (SIZEOF(TYPEOF(cre\component_feature.associated_component) *
        ['AP210_ARM.CONTACT_SIZE_DEPENDENT_LAND',
          'AP210_ARM.' +
            'COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_LAND',
            'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND']) = 1))
      )) = 0;
  WR3: SIZEOF(QUERY(cre &#60;* connectivity_requirement_element |
    NOT(NOT('AP210_ARM.NON_FUNCTIONAL_LAND_JOIN_TERMINAL' IN TYPEOF(cre))
      OR (SIZEOF(TYPEOF(cre\component_feature.associated_component) *
        ['AP210_ARM.CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND',
          'AP210_ARM.' +
            'COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND',
            'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND']) = 1))
      )) = 0;

END_ENTITY; -- physical_network

( *
```

4.2.661 Physical_network_supporting_inter_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_network_supporting_inter_stratum_feature` is a type of `Inter_stratum_join_implementation` (see 4.2.436) and a type of `Plated_inter_stratum_feature` (see 4.2.687), that is included in the design for the purpose of implementing connectivity. A `Physical_network_supporting_inter_stratum_feature` references the `Physical_network` it is supporting through a relationship to the `Join_relationship` application object that it inherits from the `Inter_stratum_join_implementation` application object.

NOTE EXPRESS specification:

*)

```
ENTITY physical_network_supporting_inter_stratum_feature
  SUPERTYPE OF (join_two_physical_connectivity_definition_supporting_inter_stratum_feature
  SUBTYPE OF (plated_inter_stratum_feature,
              inter_stratum_join_implementation);
END_ENTITY; -- physical_network_supporting_inter_stratum_feature
```

(*

4.2.662 `Physical_network_supporting_stratum_feature_conductive_join`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_network_supporting_stratum_feature_conductive_join` is a type of `Inter_stratum_join_implementation` (see 4.2.436) and a type of `Stratum_feature_conductive_join` (see 4.2.853), that is included in the design for the purpose of implementing a `Physical_network`. A `Physical_network_supporting_stratum_feature_conductive_join` effects an interconnection from a succeeding conductive material deposition to a preceding conductive material deposition for the purpose of implementing connectivity.

Constraints:

WR1: Each `Stratum_feature` interconnected shall be resident on a `Stratum` that is a resident stratum for a `Layer_connection_point` specified by the topological requirement that this implementation satisfies.

WR2: Each `Stratum` referenced by a `Stratum_feature` interconnected shall be disjoint from `Stratum` referenced by any other `Stratum_feature` interconnected.

NOTE EXPRESS specification:

*)

```
ENTITY physical_network_supporting_stratum_feature_conductive_join
  SUBTYPE OF (stratum_feature_conductive_join,
              inter_stratum_join_implementation);
END_ENTITY; -- physical_network_supporting_stratum_feature_conductive_join
```

(*

4.2.663 **Physical_node_requirement_to_implementing_component_allocation**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_node_requirement_to_implementing_component_allocation` is a type of `Managed_design_object` (see 4.2.524). A `Physical_node_requirement_to_implementing_component_allocation` is the association between the connectivity requirements and the component and joints that provide that connectivity. The component specified by `implementation_component` participate in an assembly relationship with the components whose terminals are specified by the `physical_node_requirement` in order for the total connectivity specified by `physical_node_requirement` to be achieved. There may be more than one member of `physical_node_requirement_to_implementing_component_allocation` for a member of `Physical_connectivity_definition` if the nodal implementation is split among more than one component.

NOTE The `Physical_node_requirement_to_implementing_component_allocation` is a link in the traceability path between the functional definition and the geometric representation of the interconnect substrate conductive patterns.

The data associated with a `Physical_node_requirement_to_implementing_component_allocation` are the following:

- `implementation_component`;
- `physical_node_requirement`;
- `implementation_interface_joint_set`;
- `implementation_joint_set`.

4.2.663.1 **implementation_component**

The `implementation_component` specifies the role of the `Assembly_component` (see 4.2.34) for the `Physical_node_requirement_to_implementing_component_allocation`. The `implementation_component` is the component physically providing the implementation of connectivity stated by the requirements.

4.2.663.2 **physical_node_requirement**

The `physical_node_requirement` specifies the role of the `Physical_connectivity_definition` (see 4.2.652) for the `Physical_node_requirement_to_implementing_component_allocation`.

4.2.663.3 **implementation_interface_joint_set**

The `implementation_interface_joint_set` specifies the role of the `Interface_mounted_join` (see 4.2.470) for the `Physical_node_requirement_to_implementing_component_allocation`. The `implementation_interface_joint_set` identifies the joints that attach the terminals specified by the connectivity requirement to the interface features of the component that is implementing that requirement.

4.2.663.4 implementation_joint_set

The implementation_joint_set specifies the role of the Assembly_joint (see 4.2.41) for the Physical_node_requirement_to_implementing_component_allocation. The implementation_joint_set identifies the joints that attach the terminals specified by the connectivity requirement to the features of the component that is implementing that requirement. The implementation_joint_set identifies the joints that attach the terminals of interface components to the features of the component that is implementing that requirement.

Constraints:

WR1: The implementation_component, implementation_interface_joint_set, implementation_joint_set, and physical_node_requirement shall be in the same design view.

WR2: Either implementation_interface_joint_set shall be provided or implementation_joint_set shall be provided, but both shall not be provided.

WR3: Each member of implementation_joint_set shall reference a feature of implementation_component in its set of assembly_features. Each member of implementation_joint_set shall either reference a Component_terminal (see 4.2.137) that is also referenced by the attribute associated_terminals of the physical_node_requirement, or shall reference a join terminal of the implementation_component referenced by another member of Physical_node_requirement_to_implementing_component_allocation where the interface terminal is referenced by implementation_interface_joint_set of that Physical_node_requirement_to_implementing_component_allocation.

NOTE This rule reduces potential confusion caused by more than one implementation_component implementing a single binary connection between two terminals.

NOTE EXPRESS specification:

```

*)
ENTITY physical_node_requirement_to_implementing_component_allocation
  SUBTYPE OF (managed_design_object);
    physical_node_requirement      : physical_connectivity_definition;
    implementation_component        : assembly_component;
    implementation_interface_joint_set : OPTIONAL SET [1:?] OF
                                      interface_mounted_join;
    implementation_joint_set        : OPTIONAL SET [1:?] OF assembly_joint;
--tt tc1 wd12 modified
DERIVE
  implementation_connection_definition_1 : SET [0:1] OF part_connected_terminals_defini
  := get_pctd(implementation_component, implementation_joint_set);
  implementation_connection_definition_2 : SET [0:1] OF packaged_connector_terminal_rel
  := get_pctr(implementation_component, implementation_interface_joint_set);
WHERE
  WR1: EXISTS(implementation_interface_joint_set) XOR
        EXISTS(implementation_joint_set);
  WR2: (SIZEOF(implementation_connection_definition_1) +
        SIZEOF(implementation_connection_definition_2) = 1)
        OR
        (( 'AP210_ARM.INTERFACE_COMPONENT' IN
            TYPEOF(implementation_component))
          AND
          ( 'AP210_ARM.COMPONENT_INTERFACE_TERMINAL' IN
            TYPEOF(get_cit(implementation_component, implementation_joint_set))))
        OR
        ( 'AP210_ARM.ROUTED_INTERCONNECT_COMPONENT' IN
          TYPEOF(implementation_component));
  WR3: SIZEOF(QUERY (pct &#60;* physical_node_requirement.associated_terminals |
                    pct\component_feature.associated_component = implementation_component )) = 0;
END_ENTITY; -- physical_node_requirement_to_implementing_component_allocation

( *
```

4.2.664 Physical_unit

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A **Physical_unit** is a type of **Ee_product_definition** (see 4.2.323). Each **Physical_unit** is either a **Physical_unit_design_view** (see 4.2.667) or a **Physical_unit_usage_view** (see 4.2.672). A **Physical_unit** is the description of physical characteristics common to a variety of realizations of an **Ee_product** (see 4.2.319). This description is applicable to every occurrence of a physical realization in composing a higher level assembly.

The data associated with a `Physical_unit` are the following:

- `analytical_physical_unit_representation`;
- `design_requirement`;
- `physical_device_characteristic`;
- `physical_device_coordinated_characteristic`;
- `unit_technology`.

4.2.664.1 analytical_physical_unit_representation

The `analytical_physical_unit_representation` specifies the `Analytical_representation` (see 4.2.22) for the instance of a computer interpretable model (including assigned parameter values) of the `Physical_unit`. The `analytical_physical_unit_representation` need not be specified for a particular subtype of `Physical_unit`. There may be more than one `Analytical_representation` for a `Physical_unit`.

NOTE Computer interpretable models are used to predict temperatures, stress, and failure rate. This is a representation because all declared parameters have values assigned so that when interpreted by a computer a report may be generated that describes the behaviour of the item represented. This report may be a graph or table.

4.2.664.2 design_requirement

The `design_requirement` specifies the `Ee_requirement_occurrence` (see 4.2.330) for the technical requirement for the `Physical_unit`. The `design_requirement` need not be specified for a particular `Physical_unit`. There may be more than one `Ee_requirement_occurrence` for a `Physical_unit`.

4.2.664.3 physical_device_characteristic

The `physical_device_characteristic` specifies the role of the `Characteristic` (see 4.2.98) for the `Physical_unit`. The `physical_device_characteristic` need not be specified for a particular `Physical_unit`. There may be more than one `Characteristic` for a `Physical_unit`.

4.2.664.4 physical_device_coordinated_characteristic

The `physical_device_coordinated_characteristic` specifies the role of the `Coordinated_characteristic` (see 4.2.197) for the `Physical_unit`. The `physical_device_coordinated_characteristic` need not be specified for a particular `Physical_unit`. There may be more than one `Coordinated_characteristic` for a `Physical_unit`.

4.2.664.5 unit_technology

The `unit_technology` specifies the `Technology` (see 4.2.886) used to realize the `Physical_unit`. The `unit_technology` need not be specified for a particular `Physical_unit`.

EXAMPLE 1 Examples of the unit_technologies for the Physical_units that are Physical_unit_design_views called Interconnect_modules include rigid, flexible, and flex-rigid multilayer printed circuit boards, and three-dimensional Molded Interconnect Devices. Another categorization of such Physical_units includes digital, analogue, mixed, and rf technologies.

EXAMPLE 2 Some examples of the unit_technologies for Physical_units that are Physical_unit_usage_views called Packages include ceramic Dual Inline (DIP), plastic DIP, Single Inline (SIP), J-Lead, metal can, ball grid array, molded axial, and molded radial packages.

EXAMPLE 3 Examples of the unit_technologies for the Physical_units that are Physical_unit_usage_views called Bare_die include CMOS, ECL, GAAS, BICMOS.

EXAMPLE 4 Examples of the unit_technologies for the Physical_units that are Physical_unit_design_views called Assembly_modules are MCM-D (Multichip Modules, Deposited), MCM-L (Multichip Modules, Laminated), MCM-C (Multichip Modules, Ceramic), and hybrid microcircuits.

NOTE For interoperability purposes, the technology data also may be included in a more generic manner in Ee_product_category.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit
  ABSTRACT SUPERTYPE OF (ONEOF (physical_unit_design_view,
    physical_unit_usage_view))
  SUBTYPE OF (ee_product_definition);
    analytical_physical_unit_representation : OPTIONAL SET [1:?] OF
                                              analytical_representation;
    unit_technology                        : OPTIONAL technology;
    physical_device_characteristic         : OPTIONAL SET [1:?] OF
                                              characteristic;
    physical_device_coordinated_characteristic : OPTIONAL SET [1:?] OF
                                              coordinated_characteristic;
    design_requirement                    : OPTIONAL SET [1:?] OF
                                              ee_requirement_occurrence;
END_ENTITY; -- physical_unit

( *
```

4.2.665 Physical_unit_3d_keepout_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_unit_3d_keepout_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Non_feature_shape_definition (see 4.2.559). A Physical_unit_3d_keepout_shape is the shape in three orthogonal dimensions of a keepout constraint for a Physical_unit (see 4.2.664).

NOTE Keepout regions typically include some minimal spacing around the extreme shape.

The data associated with a `Physical_unit_3d_keepout_shape` are the following:

- `application_technology_constraint`;
- `centroid_location`;
- `constrained_design_object_category`;
- `shape_characterized_physical_unit`;
- `shape_environment`;
- `shape_material_condition`;
- `shape_purpose`;
- `shape_representation`.

4.2.665.1 application_technology_constraint

The `application_technology_constraint` specifies the role of `Ee_requirement_occurrence` for the `Physical_unit_3d_keepout_shape`. The `application_technology_constraint` is the relationship between keepout shapes and process constraints under which those keepout shapes are valid.

EXAMPLE 1 An example of an `application_technology_constraint` is the relationship of the shape of a clearance area to an assembly or testing process which requires the clearance.

4.2.665.2 centroid_location

The `centroid_location` is a `Cartesian_point` (see 4.2.87) which is the center of a `Physical_unit_3d_keepout_shape`. A `centroid_location` need not be specified for a particular `Physical_unit_3d_keepout_shape`.

4.2.665.3 constrained_design_object_category

The `constrained_design_object_category` specifies the category for which the `Physical_unit_3d_keepout_shape` is a constraint.

4.2.665.4 shape_characterized_physical_unit

The `shape_characterized_physical_unit` specifies the role of the `Physical_unit` (see 4.2.664) for the `Physical_unit_3d_keepout_shape`.

4.2.665.5 shape_environment

The `shape_environment` specifies either manufacturing or an `end_user_application` as the environment for the `Physical_unit_3d_keepout_shape`.

4.2.665.6 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Physical_unit_3d_keeput_shape.

4.2.665.7 shape_purpose

The shape_purpose specifies either electro_magnetic_compatibility, shock, thermal, vibration, or an external_definition as the purpose for the Physical_unit_3d_keeput_shape.

4.2.665.8 shape_representation

The shape_representation specifies the role of the Bound_volume_shape (see 4.2.76) for the Physical_unit_3d_keeput_shape.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_3d_keeput_shape
  SUBTYPE OF (bound_volume_shape, non_feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    constrained_design_object_category : keeput_product_design_object_category;
    shape_characterized_physical_unit : physical_unit;
    shape_material_condition          : material_condition;
    shape_purpose                      : keeput_purpose;
    shape_environment                 : environment;
    centroid_location                 : OPTIONAL cartesian_point;
END_ENTITY; -- physical_unit_3d_keeput_shape
( *
```

4.2.666 Physical_unit_3d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_unit_3d_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Physical_unit_3d_shape is the shape in three orthogonal dimensions of a Physical_unit (see 4.2.664). A Physical_unit_3d_shape may be included for purposes which include the actual shape or boundaries of the actual shape. A Physical_unit (see 4.2.664) shall not be referenced by a Physical_unit_3d_shape.

NOTE 1 When a extreme boundary of the actual shape is provided, mass property calculations based on the extreme shape may be invalid, depending on the purpose. For example, the centroid_location in particular may not be valid for an extreme shape representing the envelope of a mult-mode vibration since the centroid may also vary over a range.

NOTE 2 The shape for a Packaged_part is provided by a Package, eliminating the requirement to support a shape for a Packaged_part separately.

The data associated with a Physical_unit_3d_shape are the following:

- application_technology_constraint;
- centroid_location;
- shape_characterized_physical_unit;
- shape_environment;
- shape_material_condition;
- shape_purpose;
- shape_representation.

4.2.666.1 application_technology_constraint

The application_technology_constraint specifies the role of Ee_requirement_occurrence for the Physical_unit_3d_shape. The application_technology_constraint is the relationship between physical shapes and process constraints under which those physical shapes are valid.

4.2.666.2 centroid_location

The centroid_location is a Cartesian_point (see 4.2.87) which is the center of a Physical_unit_3d_shape. A centroid_location need not be specified for a particular Physical_unit_3d_shape.

4.2.666.3 shape_characterized_physical_unit

The shape_characterized_physical_unit specifies the role of the Physical_unit (see 4.2.664) for the Physical_unit_3d_shape.

4.2.666.4 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the environment for the Physical_unit_3d_shape.

4.2.666.5 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Physical_unit_3d_shape.

4.2.666.6 shape_purpose

The shape_purpose specifies either a design, a design_profile, a design_profile_above_seating_plane, a design_profile_below_seating_plane, an electro_magnetic_compatibility_analysis, a shock_analysis, a thermal_analysis, or a vibration_analysis as the purpose for the Physical_unit_3d_shape.

4.2.666.7 shape_representation

The shape_representation specifies the role of the Bound_volume_shape (see 4.2.76) for the Physical_unit_3d_shape.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_3d_shape
  SUBTYPE OF (bound_volume_shape, feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    shape_characterized_physical_unit : physical_unit;
    shape_material_condition          : material_condition;
    shape_purpose                      : three_d_purpose;
    shape_environment                 : environment;
    centroid_location                 : OPTIONAL cartesian_point;
WHERE
  WR1: NOT('AP210_ARM.PACKAGED_PART' IN TYPEOF(shape_characterized_physical_unit));
END_ENTITY; -- physical_unit_3d_shape
(

```

4.2.667 Physical_unit_design_view

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Physical_unit_design_view is a type of Physical_unit (see 4.2.664). A Physical_unit_design_view may be one or both of an Assembly_module (see 4.2.43) or an Interconnect_module (see 4.2.440). A Physical_unit_design_view shall fully describe a specifically designed item in sufficient detail to permit production of that item in a manufacturing process and analysis of that item's response to a simulated environment in a computer based experimental environment.

EXAMPLE 1 Figure 80 illustrates an assembly design view.

The data associated with a `Physical_unit_design_view` are the following:

— `usage_view`.

4.2.667.1 `usage_view`

The `usage_view` specifies the role of the `Physical_unit_usage_view` (see 4.2.672) for the `Physical_unit_design_view`. The `usage_view` provides sufficient detail to allow the `Physical_unit` (see 4.2.664) to be utilized in its next assembly, but does not provide sufficient detail to allow reproduction of the `Physical_unit`. The `usage_view` specifies the features visible in the next assembly and the terminals of the `Physical_unit`, if there are any. The `usage_view` shall not be supplied for the `Layout_macro_definition` but shall be supplied for all other types of `Physical_unit_design_view`.

`Pca` is a type of `Assembly_module`

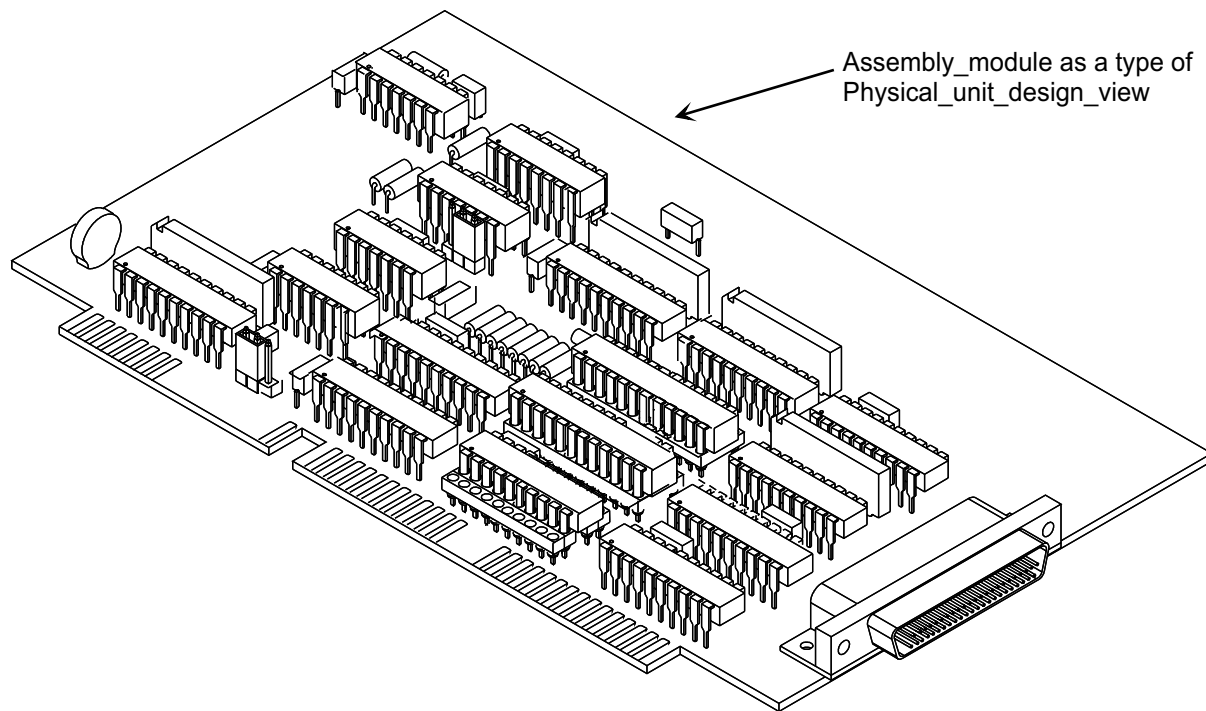


Figure 80 – `Physical_unit_design_view`

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_design_view
--tt tc1 wd11 modified
    SUPERTYPE OF (ONEOF(assembly_module,
                        interconnect_module))
    SUBTYPE OF (physical_unit);
    usage_view : OPTIONAL physical_unit_usage_view;
WHERE
    WR1: ( 'AP210_ARM.LAYOUT_MACRO_DEFINITION' IN TYPEOF(SELF)) OR
    EXISTS(usage_view);
END_ENTITY; -- physical_unit_design_view

( *
```

4.2.668 Physical_unit_network_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_unit_network_definition` is a type of `Ee_product_definition` (see 4.2.323). A `Physical_unit_network_definition` specifies a network of physical nodes, each of which references several physical terminals. Each node in the network is isolated from the other nodes in the network and is separately identifiable.

NOTE This Application object corresponds to the `Functional_unit_network_definition` but the nodes and terminals in that AO are abstract.

EXAMPLE 1 A `Physical_unit_network_definition` is referenced by members of `Physical_connectivity_definition` that correspond to the connectivity input to a design process for interconnect design.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_network_definition
    SUBTYPE OF (ee_product_definition);
END_ENTITY; -- physical_unit_network_definition

( *
```

4.2.669 Physical_unit_planar_keepout_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_unit_planar_keepout_shape` is a type of `Planar_shape` (see 4.2.678) and a type of `Non_feature_shape_definition` (see 4.2.559). A `Physical_unit_planar_keepout_shape` is the projected shape of a keepout region around the `Physical_unit` (see 4.2.664) in two dimensions.

NOTE 1 Keepout regions typically include some minimal spacing around the extreme shape represented by `Physical_unit_planar_shape` where the exact relationship between the two shapes is unrecorded.

NOTE 2 There may be multiple `Physical_unit_planar_keepout_shape` for a particular purpose. The recommended method to aggregate several `Physical_unit_planar_keepout_shape` that together describe a volumetric constraint is to reference them from a member of `Shape_and_ee_product_definition_based_constraint_occurrence` (see 4.2.806).

The data associated with a `Physical_unit_planar_keepout_shape` are the following:

- `application_technology_constraint`;
- `centroid_location`;
- `constrained_design_object_category`;
- `shape_characterized_physical_unit`;
- `shape_environment`;
- `shape_distance_from_seating_plane`;
- `shape_location_with_respect_to_seating_plane`;
- `shape_material_condition`;
- `shape_purpose`.

4.2.669.1 application_technology_constraint

The `application_technology_constraint` specifies the role of `Ee_requirement_occurrence` for the `Physical_unit_3d_shape`. The `application_technology_constraint` is the relationship between keepout shapes and process constraints under which those keepout shapes are valid.

EXAMPLE 1 An example of an `application_technology_constraint` is the relationship of the shape of a clearance area to an assembly or testing process which requires the clearance.

4.2.669.2 centroid_location

The `centroid_location` is a `Cartesian_point` (see 4.2.87) which is the center of a `Physical_unit_planar_keepout_shape`. A `centroid_location` need not be specified for a particular `Physical_unit_planar_keepout_shape`.

4.2.669.3 constrained_design_object_category

The `constrained_design_object_category` specifies the category of design object for which the `Physical_unit_planar_keepout_shape` is a constraint.

4.2.669.4 shape_characterized_physical_unit

The shape_characterized_part_unit specifies the role of the Physical_unit (see 4.2.664) for the Physical_unit_planar_keepout_shape.

4.2.669.5 shape_environment

The shape_environment specifies either manufacturing or an end_user_application as the environment for the Physical_unit_planar_keepout_shape.

4.2.669.6 shape_distance_from_seating_plane

The shape_distance_from_seating_plane specifies the distance of the Physical_unit_planar_keepout_shape from the seating_plane. The shape_distance_from_seating_plane is coordinated with the shape_location_with_respect_to_seating_plane. The shape_distance_from_seating_plane need not be provided with a Physical_unit_planar_keepout_shape.

4.2.669.7 shape_location_with_respect_to_seating_plane

The shape_location_with_respect_to_seating_plane specifies the location of the Physical_unit_planar_keepout_shape as 'above', 'below' or 'congruent' with the seating_plane.

4.2.669.8 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Physical_unit_planar_keepout_shape.

4.2.669.9 shape_purpose

The shape_purpose specifies either electro_magnetic_compatibility, shock, thermal, vibration, or an external_definition as the purpose for the Physical_unit_planar_keepout_shape.

Constraints:

WR1: If the shape_location_with_respect_to_seating_plane value is 'congruent', the shape_distance_from_seating_plane shall not be supplied.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_planar_keepout_shape
  SUBTYPE OF (planar_shape, non_feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    constrained_design_object_category : keepout_product_design_object_category;
    shape_characterized_physical_unit : physical_unit;
    shape_distance_from_seating_plane : OPTIONAL length_data_element;
    shape_location_with_respect_to_seating_plane: shape_location;
    shape_material_condition          : material_condition;
    centroid_location                 : OPTIONAL cartesian_point;
    shape_environment                 : environment;
    shape_purpose                       : keepout_purpose;
WHERE
  WR1: NOT((congruent =
    shape_location_with_respect_to_seating_plane) AND
    (EXISTS(shape_distance_from_seating_plane)));
END_ENTITY; -- physical_unit_planar_keepout_shape
( *

```

4.2.670 Physical_unit_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_unit_planar_shape` is a type of `Planar_projected_shape` (see 4.2.677). A `Physical_unit_planar_shape` is the projected shape of the `Physical_unit` (see 4.2.664) in two dimensions. A `Physical_unit_planar_shape` shall not reference a `Packaged_part` (see 4.2.591).

NOTE 1 When an extreme boundary of the actual shape is provided, static mass property calculations based on the extreme shape may be invalid, depending on the purpose.

NOTE 2 The shape of the Package is the shape for a `Packaged_part` that references that Package, eliminating the requirement to provide a shape for a `Packaged_part`.

The data associated with a `Physical_unit_planar_shape` are the following:

- `application_technology_constraint`;
- `centroid_location`;
- `shape_characterized_physical_unit`;
- `shape_distance_from_seating_plane`;
- `shape_environment`;
- `shape_location_with_respect_to_seating_plane`;
- `shape_material_condition`;
- `shape_purpose`.

4.2.670.1 `application_technology_constraint`

The `application_technology_constraint` specifies the role of `Ee_requirement_occurrence` for the `Physical_unit_3d_shape`. The `application_technology_constraint` is the relationship between physical shapes and process constraints under which those shapes are valid.

4.2.670.2 `centroid_location`

The `centroid_location` is a `Cartesian_point` (see 4.2.87) which is the center of a `Physical_unit_planar_shape`. A `centroid_location` need not be specified for a particular `Physical_unit_planar_shape`.

4.2.670.3 `shape_characterized_physical_unit`

The `shape_characterized_part_unit` specifies the role of the `Physical_unit` (see 4.2.664) for the `Physical_unit_planar_shape`.

4.2.670.4 `shape_distance_from_seating_plane`

The `shape_distance_from_seating_plane` specifies the distance of the `Physical_unit_planar_shape` from the seating plane. The `shape_distance_from_seating_plane` is coordinated with the `shape_location_with_respect_to_seating_plane`. The `shape_distance_from_seating_plane` need not be provided with a `Physical_unit_planar_shape`.

4.2.670.5 `shape_environment`

The `shape_environment` specifies either manufacturing or an `end_user_application` as the environment for the `Physical_unit_planar_shape`.

4.2.670.6 shape_location_with_respect_to_seating_plane

The shape_location_with_respect_to_seating_plane specifies the location of the Physical_unit_planar_shape as 'above', 'below' or 'congruent' with the seating_plane.

4.2.670.7 shape_material_condition

The shape_material_condition specifies either the maximum_material_condition, the minimum_material_condition, or the nominal_material_condition as the material condition for the Physical_unit_planar_shape.

4.2.670.8 shape_purpose

The shape_purpose specifies either a design, a design_profile, a design_profile_above_seating_plane, a design_profile_below_seating_plane, an electro_magnetic_compatibility_analysis, a shock_analysis, a thermal_analysis, or a vibration_analysis, as the purpose for the Physical_unit_planar_shape.

Constraints:

WR1: If the shape_characterized_physical_unit attribute specifies a Bare_die, or Package, and the shape_environment is equal to "manufacturing" and the shape_purpose is equal to "design" then the planar shape shall be oriented parallel to the seating plane and the distance to the seating plane shall be greater than any point in the volume of the Physical_unit.

NOTE This rule provides a shape representation useful for the top view of an item as may be seen in an assembly.

NOTE EXPRESS specification:

```

*)
ENTITY physical_unit_planar_shape
  SUBTYPE OF (planar_projected_shape);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    shape_characterized_physical_unit : physical_unit;
    shape_distance_from_seating_plane : OPTIONAL length_data_element;
    shape_location_with_respect_to_seating_plane: shape_location;
    shape_material_condition          : material_condition;
    centroid_location                 : OPTIONAL cartesian_point;
    shape_environment                 : environment;
    shape_purpose                       : planar_purpose;
WHERE
  WR1: NOT('AP210_ARM.PACKAGED_PART' IN TYPEOF(shape_characterized_physical_unit));
END_ENTITY; -- physical_unit_planar_shape
( *
```

4.2.671 Physical_unit_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_unit_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Physical_unit_relationship` is the association of two `Physical_unit` (see 4.2.664).

The data associated with a `Physical_unit_relationship` are the following:

- `related_unit`;
- `relating_unit`.

4.2.671.1 `related_unit`

The `related_unit` specifies the dependent `Physical_unit` (see 4.2.664) that is related to the independent `Physical_unit` in a `Physical_unit_relationship`.

4.2.671.2 `relating_unit`

The `relating_unit` specifies the independent `Physical_unit` (see 4.2.664) to which the `Physical_unit_relationship` is made.

NOTE EXPRESS specification:

```
*)
ENTITY physical_unit_relationship
  SUBTYPE OF (managed_design_object);
    relating_unit : physical_unit;
    related_unit  : physical_unit;
END_ENTITY; -- physical_unit_relationship

( *
```

4.2.672 `Physical_unit_usage_view`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Physical_unit_usage_view` is a type of `Physical_unit` (see 4.2.664). Each `Physical_unit_usage_view` may be one of an `Assembly_module_usage_view` (see 4.2.51), a `Bare_die` (see 4.2.59), an `Interconnect_module_usage_view` (see 4.2.465), a `Package` (see 4.2.573), or a `Packaged_part` (see 4.2.591). A `Physical_unit_usage_view` identifies a `Physical_unit` as a part in its next assembly with sufficient detail to allow its use but without sufficient detail to permit reproduction of the `Physical_unit`. The product version referenced by the version attribute inherited from `Ee_product_definition` need not reference the same product version referenced by the design view.

NOTE 1 The product information necessary for a `Physical_unit_usage_view` is much less than that necessary to design, manufacture, or verify its performance from first principles. Terminals are defined only in the usage view of an item in this part of ISO 10303.

NOTE 2 The usage view and design view of product contain similar concepts. It is the responsibility of the design organization to create the mapping between the usage view and the design view. As the design evolves to meet the requirements, the design view will reference different versions than the usage view.

NOTE 3 In a drawing based exchange, an interface control drawing would contain the same information found in a usage view. Industry standards describing the information represented by interface control drawings could migrate to computer interpretable standards exchange using this standard.

EXAMPLE 1 Examples of information provided in a usage view include outline shape, functional block diagram, terminal identification, package orientation features, mounting features, interface models, pin mapping.

The data associated with a `Physical_unit_usage_view` are the following:

— `approved_part`.

4.2.672.1 `approved_part`

The `approved_part` specifies the `Supplied_product_version` (see 4.2.879) that satisfies the intended application of a certain `Physical_unit_usage_view`. The `approved_part` need not be specified for a particular `Physical_unit_usage_view`. There may be more than one `Supplied_product_version` for a `Physical_unit_usage_view`.

NOTE EXPRESS specification:

```
*)
ENTITY physical_unit_usage_view
  SUPERTYPE OF (ONEOF (package, bare_die, packaged_part,
    assembly_module_usage_view, interconnect_module_usage_view))
  SUBTYPE OF (physical_unit);
-- tt deleted aug 25, 2001
-- material : OPTIONAL SET [1:?] OF ee_material;
END_ENTITY; -- physical_unit_usage_view

(*
```

4.2.673 `Pitch_class`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pitch_class is a type of Managed_design_object (see 4.2.524). A Pitch_class is a category in which a particular group of Part_feature (see 4.2.609) may fall, depending on the value of their paired spacing. The Pitch_class controls the width of the mating feature which is measured perpendicular to the primary axis of a terminal in the class. The only Data_element type permitted in the range is a Length_data_element.

EXAMPLE 1 The Pitch_class is included so that tolerances may be applied to all members of the same class, and specified only once. E.g., all pitches with a design spacing greater than 50 mils and less than 75 mils are in the 'big' class.

NOTE Other Application Objects will be populated in order to apply the class property to tolerances of the product being designed .

The data associated with a Pitch_class are the following:

- class_member;
- class_name.

4.2.673.1 class_member

The class_member specifies the role of the Range_characteristic for the Pitch_class. The range specifies the spacings which are included as a member of the Pitch_class. The Range_characteristic shall use Length_data_element for the Numeric_parameter.

4.2.673.2 class_name

The class_name specifies the name for the Pitch_class.

NOTE EXPRESS specification:

```

*)
ENTITY pitch_class
  SUBTYPE OF (managed_design_object);
    class_member : range_characteristic;
    class_name   : ee_name;
WHERE
  WR1: SIZEOF(QUERY( np &#60;* class_member\coordinated_characteristic.component_characteri
    NOT('AP210_ARM.LENGTH_DATA_ELEMENT' IN TYPEOF(np\characteristic.property_value))
  )) = 0;
END_ENTITY; -- pitch_class

( *
```

4.2.674 Placement_group

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Placement_group is a type of Managed_design_object (see 4.2.524). A Placement_group is a group composed of Assembly_component (see 4.2.34) that are associated with each other for this purpose by the Component_group_assignment (see 4.2.124), in order to meet some specific placement requirement. The placement is assigned through the Placement_group_area_assignment (see 4.2.675) object, and does not indicate the explicit geometric relationship of the Assembly_components to each other, only that they are all to be placed within an area.

EXAMPLE 1 An example of a Placement_group is the collection of Assembly_components necessary to satisfy the functional requirement for a power supply.

EXAMPLE 2 An example of a Placement_group is the collection of Assembly_components necessary to satisfy the functional requirement for a digital signal processor.

EXAMPLE 3 An example of a Placement_group is the collection of Assembly_components with a case temperature greater than 55 degrees Centigrade. The associated area would be a rectangle all of whose inner points are within 6 cm of the cooling unit.

The data associated with a Placement_group are the following:

- design_specific_usage;
- requirement.

4.2.674.1 design_specific_usage

The design_specific_usage specifies the name of the Placement_group.

EXAMPLE 1 Examples of design_specific_usage include analogue input buffer, preprocessor, emi filter, video multiplexor, and high speed digital.

4.2.674.2 requirement

The requirement specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Placement_group. There shall be one or more Ee_requirement_occurrence for a Placement_group.

NOTE EXPRESS specification:

```
*)
ENTITY placement_group
  SUBTYPE OF (managed_design_object);
  design_specific_usage : ee_name;
  requirement           : SET [1:?] OF ee_requirement_occurrence;
END_ENTITY; -- placement_group

( *
```

4.2.675 Placement_group_area_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Placement_group_area_assignment is a type of Managed_design_object (see 4.2.524). A Placement_group_area_assignment is the relationship between a Placement_group (see 4.2.674) and the Mounting_restriction_area (see 4.2.550) on the interconnect substrate instance to which it is restricted.

The data associated with a Placement_group_area_assignment are the following:

- area;
- placed_group.

4.2.675.1 area

The area specifies the role of the Mounting_restriction_area (see 4.2.550) for the Placement_group_area_assignment.

4.2.675.2 placed_group

The placed_group specifies the role of the Placement_group (see 4.2.674) for the Placement_group_area_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY placement_group_area_assignment
  SUBTYPE OF (managed_design_object);
    placed_group : placement_group;
    area         : mounting_restriction_area;
END_ENTITY; -- placement_group_area_assignment

--tt tc1 wd9 modified
( *
```

4.2.676 Placement_group_volume_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Placement_group_volume_assignment is a type of Managed_design_object (see 4.2.524). A Placement_group_volume_assignment is the relationship between a Placement_group (see 4.2.674) and the Mounting_restriction_volume (see 4.2.551) on the interconnect substrate instance to which it is restricted.

The data associated with a Placement_group_volume_assignment are the following:

- volume;
- placed_group.

4.2.676.1 volume

The volume specifies the role of the Mounting_restriction_volume (see 4.2.551) for the Placement_group_volume_assignment.

4.2.676.2 placed_group

The placed_group specifies the role of the Placement_group (see 4.2.674) for the Placement_group_volume_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY placement_group_volume_assignment
  SUBTYPE OF (managed_design_object);
    placed_group      : placement_group;
    volume            : mounting_restriction_volume;
END_ENTITY; -- placement_group_volume_assignment

--tt tc1 wd9 modified
( *
```

4.2.677 Planar_projected_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Planar_projected_shape is a type of Planar_shape (see 4.2.678). A Planar_projected_shape may be one of an Assembly_component_2d_shape (see 4.2.35), a Design_intent_modification_planar_shape (see 4.2.272), a Part_template_planar_shape (see 4.2.628), a Physical_unit_planar_shape (see 4.2.670), a Stratum_feature_planar_shape (see 4.2.857), or a Stratum_planar_shape (see 4.2.867). A Planar_projected_shape is also a type of Feature_shape_definition (see 4.2.372). A Planar_projected_shape is projected from a solid. A Planar_projected_shape that is an Assembly_component_2d_shape may also be either a Part_template_planar_shape or a Physical_unit_planar_shape.

NOTE A Planar_shape may also be projected from a solid to handle the case where a Non_feature_shape_definition is needed that is not predefined in this standard. The predefined non feature shape projections included in this standard are Physical_unit_3d_keepout_shape (see 4.2.665) and Physical_unit_planar_keepout_shape (see 4.2.669).

The data associated with a `Planar_projected_shape` are the following:

— `curve_basis`.

4.2.677.1 `curve_basis`

The `curve_basis` is an inverse relationship that specifies `Volume_shape_projection` (see ??) that specifies the `Planar_projected_shape` as its `surface_projection`. There shall be no more than one `Volume_shape_projection` for a `Planar_projected_shape`.

NOTE EXPRESS specification:

```

*)
ENTITY planar_projected_shape
  SUPERTYPE OF (ONEOF(
    (assembly_component_2d_shape ANDOR (ONEOF(
      part_template_planar_shape,
      footprint_definition_shape,
      padstack_definition_shape,
      physical_unit_planar_shape))),
    design_intent_modification_planar_shape,
    stratum_feature_planar_shape,
    stratum_planar_shape))
  SUBTYPE OF (planar_shape, feature_shape_definition);
  INVERSE
    curve_basis_1 : SET [0:1] OF volume_shape_plane_projection FOR surface_projection_result;
    curve_basis_2 : SET [0:1] OF volume_shape_intersection FOR surface_intersection_result;
  WHERE
    WR1: EXISTS (curve_basis_1) XOR EXISTS (curve_basis_2);
END_ENTITY; -- planar_projected_shape
(
  *
```

4.2.678 `Planar_shape`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Planar_shape` is a type of `Shape_definition` (see 4.2.808) that lies on a two-dimensional plane. A `Planar_shape` may be one of a `geometrically_bounded_2d_wireframe` (see 4.2.424), a `shell_based_2d_wireframe` (see 4.2.819), an `edge_based_2d_wireframe` (see 4.2.309), or a `csg_2d_shape` (see 4.2.200). A `Planar_shape` may also be either a `Planar_projected_shape` or a `Physical_unit_planar_keepout_shape`.

NOTE A `Planar_projected_shape` is a physical shape, while a `Physical_unit_planar_keepout_shape` is a keepout region constraint shape. The Application Object name for `Planar_projected_shape` is consistent with the previous edition of this standard.

NOTE EXPRESS specification:

```

*)
ENTITY planar_shape
  ABSTRACT SUPERTYPE OF (ONEOF(
    geometrically_bounded_2d_wireframe,
    shell_based_2d_wireframe,
    edge_based_2d_wireframe,
    csg_2d_shape ) ANDOR
    (ONEOF(planar_projected_shape,
      physical_unit_planar_keepout_shape)))
  SUBTYPE OF (shape_definition);
WHERE
  WR1 : SELF\shape_definition.geometric_context.dimensionality
      = two_dimensional;
END_ENTITY; -- planar_shape

( *

```

4.2.679 Planned_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Planned_characteristic` is a type of `Managed_design_object` (see 4.2.524). A `Planned_characteristic` associates design intent information based on manufacturing capabilities and component characteristics with an allocated requirement. A `Planned_characteristic` is a set of properties planned to be valid for all occurrences of the designed product under the conditions defined in the allocated requirement.

The data associated with a `Planned_characteristic` are the following:

- `planned_coordinated_characteristic`;
- `reference_requirement`.

4.2.679.1 planned_coordinated_characteristic

The `planned_coordinated_characteristic` specifies the role of the `Coordinated_characteristic` (see 4.2.197) for the `Planned_characteristic`.

4.2.679.2 reference_requirement

The `reference_requirement` specifies the `Assembly_requirement_allocation` (see 4.2.53), `Functional_unit_requirement_allocation` (see 4.2.406), or the `Requirement_allocation` (see 4.2.769) for the `Planned_characteristic`.

NOTE This information is part of the design view for archival purposes of the design organization.

NOTE EXPRESS specification:

```

*)
ENTITY planned_characteristic
  SUBTYPE OF (managed_design_object);
    reference_requirement      : requirement_allocation_select;
    planned_coordinated_characteristic : coordinated_characteristic;
END_ENTITY; -- planned_characteristic

( *
```

4.2.680 Planned_date_effectivity

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Planned_date_effectivity is a type of Planned_effectivity (see 4.2.681). A Planned_date_effectivity is the specification by the design organization of the intended use of an Ee_product (see 4.2.319) or of an Ee_product within an Ee_product_configuration (see 4.2.322), determined by one or two dates.

The data associated with a Planned_date_effectivity are the following:

- end_date;
- start_date.

4.2.680.1 end_date

The end_date specifies either the final Date (see 4.2.212) or the final Date_and_time (see 4.2.213) upon which the Ee_product (see 4.2.319) is planned by a design organization to be used in the Ee_product_configuration (see 4.2.322), as defined by the Planned_date_effectivity. The end_date need not be specified for a particular Planned_date_effectivity.

4.2.680.2 start_date

The start_date specifies either the initial Date (see 4.2.212) or the initial Date_and_time (see 4.2.213); upon which the Ee_product (see 4.2.319) is planned by a design organization to be used in the Ee_product_configuration (see 4.2.322), as defined by the Planned_date_effectivity.

NOTE EXPRESS specification:

```

*)
ENTITY planned_date_effectivity
  SUBTYPE OF (planned_effectivity);
    start_date : date_or_date_and_time;
    end_date   : OPTIONAL date_or_date_and_time;
END_ENTITY; -- planned_date_effectivity

( *

```

4.2.681 Planned_effectivity

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Planned_effectivity is a type of Managed_design_object (see 4.2.524). A Planned_effectivity may be a Planned_date_effectivity (see 4.2.680), Planned_lot_effectivity (see 4.2.682), or Planned_serial_number_effectivity (see 4.2.683). A Planned_effectivity is the specification by a design organization of the intended use of an Ee_product (see 4.2.319) or of an Ee_product within a particular Ee_product_configuration (see 4.2.322). The Planned_effectivity details the circumstances under which the usage will take place.

The data associated with a Planned_effectivity are the following:

- component_quantity;
- configuration;
- effective_assembly;
- effectivity_approval.

4.2.681.1 component_quantity

The component_quantity specifies the number of parts as a Data_element (see 4.2.211) that are included in the affected Ee_product_configuration (see 4.2.322). The component_quantity need not be specified for a particular Planned_effectivity.

4.2.681.2 configuration

The configuration specifies the Ee_product_configuration (see 4.2.322) that is effected by the Planned_effectivity.

4.2.681.3 effective_assembly

The effective_assembly specifies the role of the Assembly_composition_relationship (see 4.2.39) for the Planned_effectivity.

4.2.681.4 effectivity_approval

The effectivity_approval specifies the Ee_approval (see 4.2.314) that is associated with the Ee_product (see 4.2.319) changes having the Planned_effectivity.

Constraints:

UR1: The combination of the configuration and the effective_assembly shall be unique within a population of Planned_effectivities.

WR1: The component_quantity, if it exists, shall be an integer greater than zero.

NOTE EXPRESS specification:

```

*)
ENTITY planned_effectivity
  SUPERTYPE OF (ONEOF (planned_date_effectivity,
    planned_serial_number_effectivity,planned_lot_effectivity))
  SUBTYPE OF (managed_design_object);
  effective_assembly : assembly_composition_relationship;
  effectivity_approval : ee_approval;
  configuration : ee_product_configuration;
  component_quantity : OPTIONAL data_element;
  UNIQUE
    UR1 : effective_assembly, configuration;
END_ENTITY; -- planned_effectivity

( *
```

4.2.682 Planned_lot_effectivity

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Planned_lot_effectivity is a type of Planned_effectivity (see 4.2.681). A Planned_lot_effectivity is the specification by the design organization of the intended use of an Ee_product (see 4.2.319) or of an Ee_product within an Ee_product_configuration (see 4.2.322), where the Ee_product or the Ee_product within an Ee_product_configuration is produced as one of a group or lot.

The data associated with a Planned_lot_effectivity are the following:

- lot_number;
- lot_size.

4.2.682.1 lot_number

The lot_number specifies the string for the alphanumeric identification of the group of Ee_product (see 4.2.319) that compose a lot for the Planned_lot_effectivity.

4.2.682.2 lot_size

The lot_size specifies the quantity of Ee_product (see 4.2.319) as a Data_element (see 4.2.211) within the lot with the Planned_lot_effectivity.

Constraints:

WR1: The lot_size shall be an integer greater than zero.

NOTE EXPRESS specification:

```
*)
ENTITY planned_lot_effectivity
  SUBTYPE OF (planned_effectivity);
    lot_number : STRING;
    lot_size   : data_element;
END_ENTITY; -- planned_lot_effectivity

( *
```

4.2.683 Planned_serial_number_effectivity

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Planned_serial_number_effectivity is a type of Planned_effectivity (see 4.2.681). A Planned_serial_number_effectivity is the specification by the design organization of the intended use of an Ee_product (see 4.2.319) identified by a range of serial numbers or of an Ee_product within Ee_product_configuration (see 4.2.322) identified by a range of serial numbers.

The data associated with a Planned_serial_number_effectivity are the following:

- from_effectivity_id;
- through_effectivity_id.

4.2.683.1 from_effectivity_id

The from_effectivity_id specifies the string for the initial serial number of the range of serial numbers for the Planned_serial_number_effectivity.

4.2.683.2 through_effectivity_id

The `through_effectivity_id` specifies the string for the final serial number of the range of serial numbers for the `Planned_serial_number_effectivity`. The `through_effectivity_id` need not be specified for a particular `Planned_serial_number_effectivity`.

NOTE EXPRESS specification:

```
*)
ENTITY planned_serial_number_effectivity
  SUBTYPE OF (planned_effectivity);
    through_effectivity_id : OPTIONAL STRING;
    from_effectivity_id    : STRING;
END_ENTITY; -- planned_serial_number_effectivity

( *
```

4.2.684 Plated_conductive_base_blind_via

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Plated_conductive_base_blind_via` is a type of `Blind_via` (see 4.2.71) in which the material that composes the floor of the `Blind_via` is conductive, and in which the floor material of the `Blind_via` is contiguous with the material on the wall of the `Blind_via`. The process that is to create this conductive contiguous material is electroplating or similar.

NOTE EXPRESS specification:

```
*)
ENTITY plated_conductive_base_blind_via
  SUBTYPE OF (blind_via);
END_ENTITY; -- plated_conductive_base_blind_via

( *
```

4.2.685 Plated_cutout

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Plated_cutout` is a type of `Plated_inter_stratum_feature` (see 4.2.687). The `Plated_cutout` is an interior volume of the substrate, such as a void included for passage of a shaft, and may be plated for EMI shielding purposes. This object conveys the semantic that the interior referenced by the related shape is conductive.

Constraints:

WR1: Each member of Plated_cutout shall not form part of the outer periphery of the Interconnect_module.

NOTE EXPRESS specification:

```

*)
ENTITY plated_cutout
  SUBTYPE OF (cutout,plated_inter_stratum_feature);
END_ENTITY; -- plated_cutout

( *
```

4.2.686 Plated_cutout_edge_segment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plated_cutout_edge_segment is a type of Plated_inter_stratum_feature (see 4.2.687) that helps compose a Partially_plated_cutout (see 4.2.634). The Plated_cutout_edge_segment is a feature with a specified shape that makes up part of the edge of a Partially_plated_cutout.

The data associated with a Plated_cutout_edge_segment are the following:

- composed_cutout;
- end_vertex;
- start_vertex.

4.2.686.1 composed_cutout

The composed_cutout specifies the Partially_plated_cutout (see 4.2.634) for whose edge this Plated_cutout_edge_segment is a segment.

4.2.686.2 end_vertex

The end_vertex specifies one role of an Edge_segment_vertex for the Plated_cutout_edge_segment.

4.2.686.3 start_vertex

The start_vertex specifies one role of an Edge_segment_vertex for the Plated_cutout_edge_segment.

Constraints:

WR1: The start_vertex shall not be the end_vertex.

NOTE EXPRESS specification:

```
*)
ENTITY plated_cutout_edge_segment
  SUBTYPE OF (cutout_edge_segment, plated_inter_stratum_feature);
  SELF\cutout_edge_segment.composed_cutout : partially_plated_cutout;
END_ENTITY; -- plated_cutout_edge_segment

( *
```

4.2.687 Plated_inter_stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plated_inter_stratum_feature is a type of Inter_stratum_feature (see 4.2.432) with deposited metal on its walls that makes an electrical connection between conductive patterns on internal Design_layer_stratum (see 4.2.275), external Design_layer_stratum, or both, of a printed circuit board. A Plated_inter_stratum_feature is one of a Plated_cutout (see 4.2.685), a Plated_cutout_edge_segment (see 4.2.686), a Plated_interconnect_module_edge (see 4.2.688), a Plated_interconnect_module_edge_segment (see 4.2.689), or a Plated_passage (see 4.2.690), or additionally may be a Physical_network_supporting_inter_stratum_feature (see 4.2.661).

NOTE 1 A cross-section illustration of Plated_inter_stratum_feature that is the result of a wet chemistry process is shown in figure 81. An expanded view of one section of the feature is shown in figure 82.

NOTE 2 Plated_inter_stratum_features that completely pass through the PCB are sometimes known as plated through holes.

NOTE 3 Plated_inter_stratum_features are features that may be custom designed for this design by creating custom Inter_stratum_feature_template (see 4.2.435) shapes.

Constraints:

WR1: Each Plated_inter_stratum_feature shall reference a Passage_technology that has the attribute plated_passage set to TRUE.

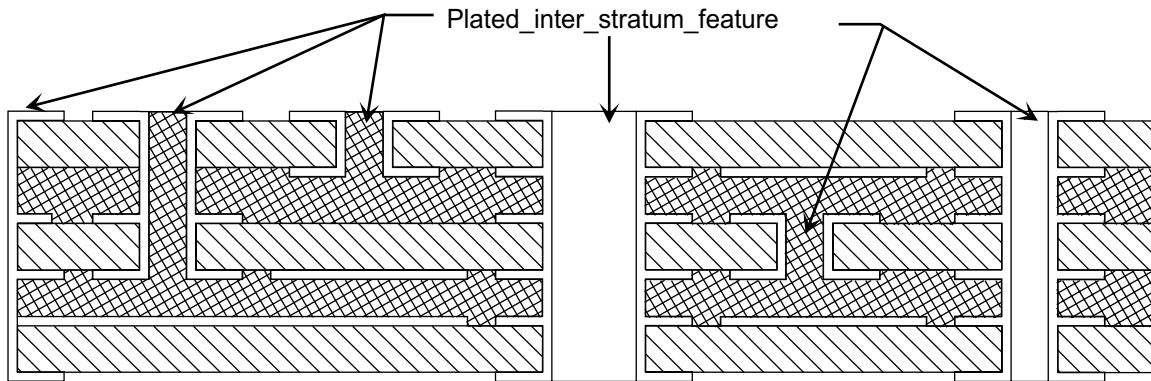


Figure 81 – Plated_inter_stratum_feature cross-section

NOTE EXPRESS specification:

```

*)
ENTITY plated_inter_stratum_feature
ABSTRACT SUPERTYPE OF (ONEOF (plated_cutout,
    plated_interconnect_module_edge, plated_cutout_edge_segment,
    plated_interconnect_module_edge_segment, plated_passage) ANDOR
    physical_network_supporting_inter_stratum_feature)
    SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- plated_inter_stratum_feature

```

(*

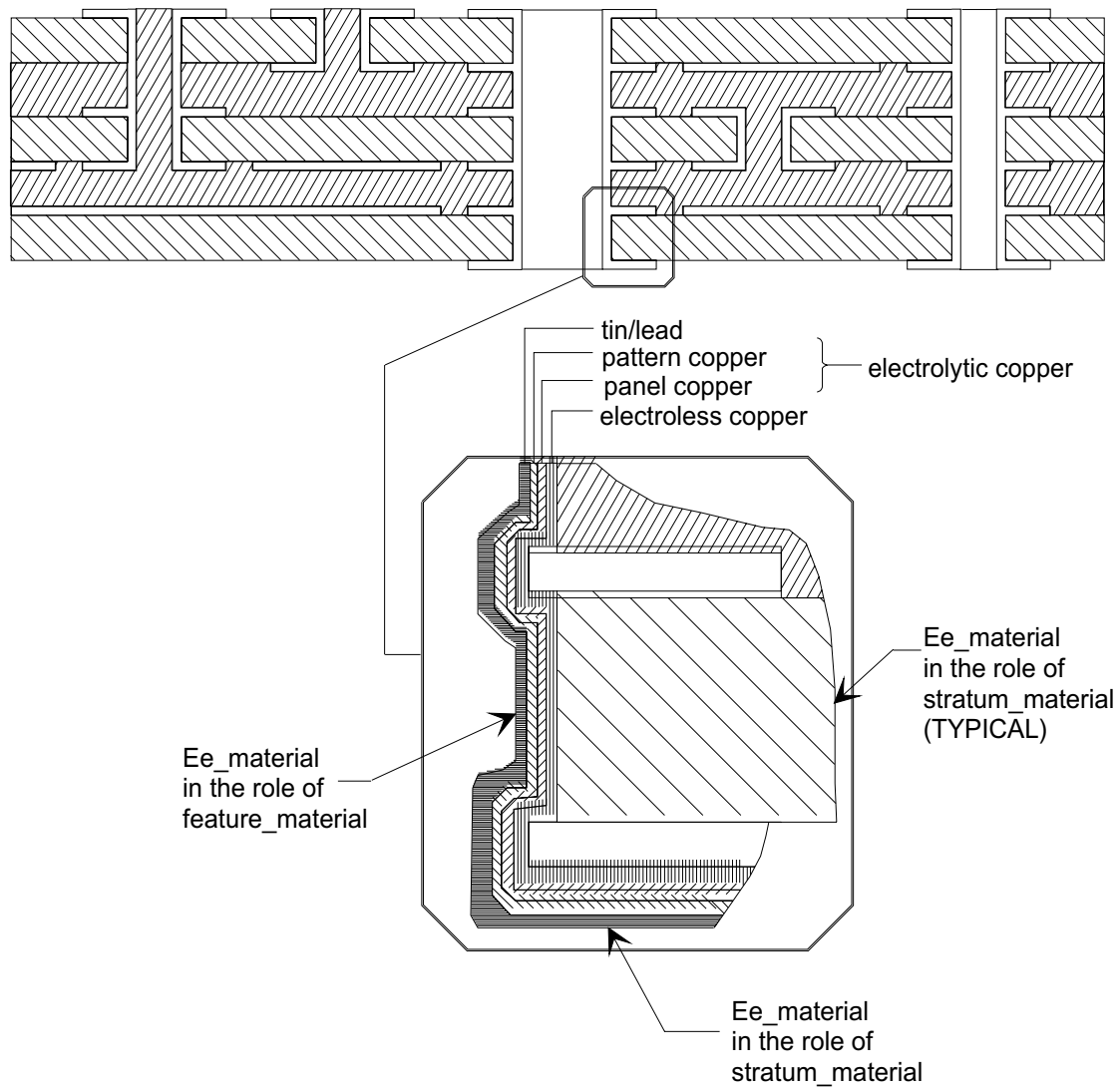
4.2.688 Plated_interconnect_module_edge

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plated_interconnect_module_edge is a type of Plated_inter_stratum_feature (see 4.2.687) that forms one of the edges of an interconnect substrate.

Constraints:

WR1: Each member of Plated_interconnect_module_edge shall only form the outer periphery of the Interconnect_module.



The Ee_material in the role of feature_material for a Plated_inter_stratum_feature is contiguous with the Ee_material in the role of stratum_material for a Stratum_technology, when the values of both Ee_material are identical.

Figure 82 – Exploded view of Plated_inter_stratum_feature

NOTE EXPRESS specification:

```

*)
ENTITY plated_interconnect_module_edge
  SUBTYPE OF (interconnect_module_edge, plated_inter_stratum_feature);
END_ENTITY; -- plated_interconnect_module_edge

( *

```

4.2.689 Plated_interconnect_module_edge_segment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plated_interconnect_module_edge_segment is a type of Plated_inter_stratum_feature (see 4.2.687) that forms a section of the edge of an interconnect substrate.

EXAMPLE 1 Figure 83 illustrates an edge segment that is plated. Note that the opposite side of the cross-section is unplated. There is not enough information in the figure to define the extent of the plated segment.

The data associated with a Plated_interconnect_module_edge_segment are the following:

- composed_edge;
- end_vertex;
- start_vertex.

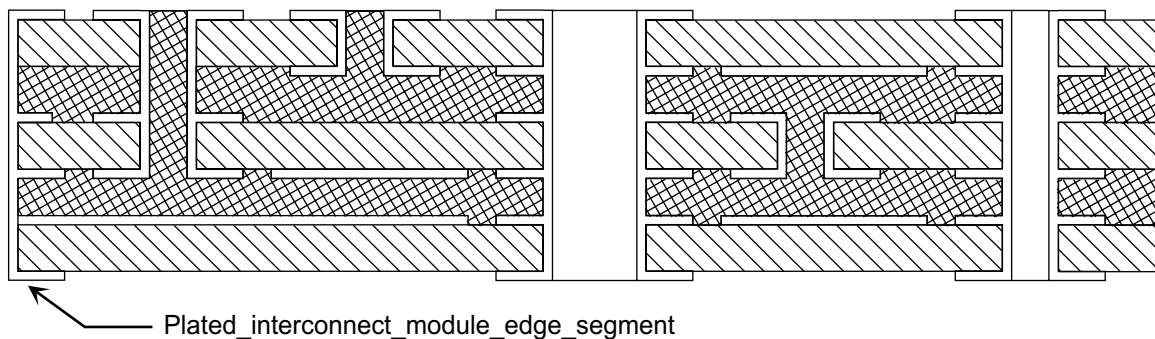


Figure 83 – Plated_interconnect_module_edge_segment

4.2.689.1 composed_edge

The composed_edge specifies the Partially_plated_interconnect_module_edge (see 4.2.635) that contains the Plated_interconnect_module_edge_segment.

4.2.689.2 end_vertex

The end_vertex specifies one role of an Edge_segment_vertex for the Plated_interconnect_module_edge_segment.

4.2.689.3 start_vertex

The start_vertex specifies one role of an Edge_segment_vertex for the Plated_interconnect_module_edge_segment.

Constraints:

WR1: The start_vertex shall not be the end_vertex.

NOTE EXPRESS specification:

```
* )
ENTITY plated_interconnect_module_edge_segment
  SUBTYPE OF (plated_inter_stratum_feature, interconnect_module_edge_segment);
  SELF\interconnect_module_edge_segment.composed_edge :
  partially_plated_interconnect_module_edge;
END_ENTITY; -- plated_interconnect_module_edge_segment

( *
```

4.2.690 Plated_passage

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plated_passage is a type of Plated_inter_stratum_feature (see 4.2.687). A Plated_passage is either a Component_termination_passage (see 4.2.143) or a Via (see 4.2.947).

EXAMPLE 1 Figure 84 graphically collects the various types of plated passages into a more generic category.

Constraints:

WR1: Each Plated_passage shall not be a member of Interconnect_module_edge.

WR2: Each Plated_passage shall not be a member of Interconnect_module_edge_segment.

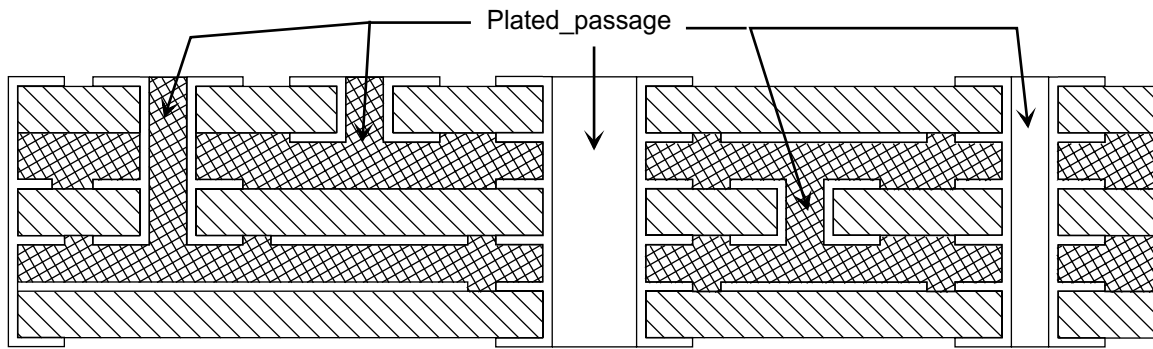


Figure 84 – Plated_passage

NOTE EXPRESS specification:

```

*)
ENTITY plated_passage
  ABSTRACT SUPERTYPE OF (ONEOF (component_termination_passage,via))
  SUBTYPE OF (plated_inter_stratum_feature);
END_ENTITY; -- plated_passage

( *

```

4.2.691 Plus_minus_tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Plus_minus_tolerance_characteristic` is a type of `Tolerance_characteristic` (see 4.2.911). The `Plus_minus_tolerance_characteristic` includes a minimum, maximum, and nominal characteristic numeric parameter. All three have the same base unit, but may have different prefixes.

The data associated with a `Plus_minus_tolerance_characteristic` are the following:

— `component_characteristic`.

4.2.691.1 component_characteristic

The `component_characteristic` specifies the nominal, minimum, and maximum values of the `Numeric_`-parameter (see 4.2.565) for the `Plus_minus_tolerance_characteristic`. There shall be one nominal, one minimum and one maximum value for the `Plus_minus_tolerance_characteristic`.

Constraints:

WR1: All Data_element referenced by Numeric_parameter in the role of component_characteristic for Plus_minus_tolerance_characteristic shall have the same unit_of_measure.

WR2: The Numeric_parameter in the role of component_characteristic shall have a type_name of minimum, nominal, or maximum.

WR3: The Numeric_parameter in the role of component_characteristic that has a type_name of minimum shall have a property_value that is less than the property_value of the Numeric_parameter in the role of component_characteristic that has a type_name of nominal.

WR4: The Numeric_parameter in the role of component_characteristic that has a type_name of nominal shall have a property_value that is less than the property_value of the Numeric_parameter in the role of component_characteristic that has a type_name of maximum.

NOTE EXPRESS specification:

```

*)
ENTITY plus_minus_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET [3:3] OF
                                                                numeric_parameter;
END_ENTITY; -- plus_minus_tolerance_characteristic

( *
```

4.2.692 Plus_minus_tolerance_range_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Plus_minus_tolerance_range_representation is a type of Shape_dimension_representation (see 4.2.809) that represents a Dimensional_characteristic (see 4.2.298) with three Data_element (see 4.2.211) from which the limits of the Dimensional_characteristic can be calculated.

The data associated with a Plus_minus_tolerance_range_representation are the following:

- basic_dimension_value;
- lower_deviation;
- upper_deviation.

4.2.692.1 basic_dimension_value

The basic_dimension_value specifies the Data_element (see 4.2.211) from which the lower and upper limits of the associated Dimensional_characteristic (see 4.2.298) may be calculated.

NOTE 1 The `basic_dimension_value` corresponds to the basic size component of toleranced dimensions and is described in ISO 286-1 and ISO 406.

NOTE 2 The `basic_dimension_value` should not be confused with basic dimensions defined in 1.3.9 of ASME Y14.5M.

4.2.692.2 lower_deviation

The `lower_deviation` specifies the `Data_element` (see 4.2.211) that is added to the `Data_element` specified by the `basic_dimension_value` to determine the lower limit of the associated `Dimensional_characteristic` (see 4.2.298).

4.2.692.3 upper_deviation

The `upper_deviation` specifies the `Data_element` (see 4.2.211) that is added to the `Data_element` specified by the `basic_dimension_value` to determine the upper limit of the associated `Dimensional_characteristic` (see 4.2.298).

Constraints:

WR1: The units of the `Data_elements` specified by the `basic_dimension_value`, `lower_deviation`, and `upper_deviation` shall be the same.

WR2: The value of the `Data_element` specified by the `lower_deviation` shall be less than the value of the `Data_element` specified by the `upper_deviation`.

NOTE EXPRESS specification:

```
*)
ENTITY plus_minus_tolerance_range_representation
  SUBTYPE OF (shape_dimension_representation);
    basic_dimension_value : data_element;
    upper_deviation       : data_element;
    lower_deviation       : data_element;
END_ENTITY; -- plus_minus_tolerance_range_representation

( *
```

4.2.693 Polarity_indication_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Polarity_indication_feature` is a type of `Part_feature` (see 4.2.609). A `Polarity_indication_feature` provides visible or tactile means to reduce or eliminate ambiguity of the location of a polarized part in an assembly operation. A `Polarity_indication_feature` shall not be used to identify orientation features of non-polarized components.

NOTE Other orientation information may be required in addition to the `Polarity_indication_feature` to totally disambiguate the correct orientation.

NOTE EXPRESS specification:

```
*)
ENTITY polarity_indication_feature
  SUBTYPE OF (part_feature);
    associated_body_vertical_extent : OPTIONAL SET [1:2] OF package_body_surface;
    associated_terminal              : package_terminal;
END_ENTITY; -- polarity_indication_feature
```

--tt tc1 correction

(*

4.2.694 Polygonal_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Polygonal_area` is a type of `Primitive_2d` (see 4.2.720).

The data associated with a `Polygonal_area` are the following:

— bounds.

4.2.694.1 bounds

The bounds specifies the role of a `Cartesian_point` (see 4.2.87) for the `Polygonal_area`. There shall be three or more `Cartesian_point` for a `Polygonal_area`. No `Cartesian_point` shall be specified more than once as a bounds for a particular `Polygonal_area`.

NOTE EXPRESS specification:

```
*)
ENTITY polygonal_area
  SUBTYPE OF (primitive_2d);
    bounds: LIST [3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- polygonal_area
```

(*

4.2.695 Position_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Position_tolerance is a type of Geometric_tolerance (see 4.2.419) that specifies or partially specifies the design requirement on the location of a Shape_element (see 4.2.810) in relation to other Shape_elements, such as a Datum_system (see 4.2.236). A Position_tolerance is either a Position_tolerance_with_referenced_datum_system (see 4.2.696) or a Position_tolerance_without_referenced_datum_system (see 4.2.697). A Position_tolerance may also be a Composite_feature_relating_position_tolerance (see 4.2.158).

NOTE Position tolerance is defined in terms of tolerance zone definitions in 14.10 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Centre_axis, Centre_plane, Centre_point, or Physical_feature_or_part_template.

WR2: If the toleranced_shape_element is a Centre_plane, the Position_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

WR3: If the toleranced_shape_element is a Centre_point the Position_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition.

IP1: The form of the toleranced Shape_element shall be consistent with the form of a shape element for which the application of a position tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY position_tolerance
  ABSTRACT SUPERTYPE OF (ONEOF (
    position_tolerance_with_referenced_datum_system,
    position_tolerance_without_referenced_datum_system) ANDOR
    composite_feature_relating_position_tolerance)
  SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- position_tolerance
```

(*

4.2.696 Position_tolerance_with_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Position_tolerance_with_referenced_datum_system` is a type of `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422) and a type of `Position_tolerance` (see 4.2.695) that participates in specifying a design requirement on the location of the tolerated `Shape_element` (see 4.2.810) in relation to a `Datum_system` (see 4.2.236). A `Position_tolerance_with_referenced_datum_system` may be either a `Composite_feature_locating_position_tolerance` (see 4.2.157) or a `Boundary_based_position_tolerance` (see 4.2.77).

NOTE EXPRESS specification:

```
*)
ENTITY position_tolerance_with_referenced_datum_system
    SUPERTYPE OF (ONEOF (composite_feature_locating_position_tolerance,
        boundary_based_position_tolerance))
    SUBTYPE OF (position_tolerance,
        geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- position_tolerance_with_referenced_datum_system
```

(*

4.2.697 Position_tolerance_without_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Position_tolerance_without_referenced_datum_system` is a type of `Position_tolerance` (see 4.2.695) and a type of `Geometric_tolerance_without_referenced_datum_system` (see 4.2.423) that participates in specifying a design requirement on the location of the tolerated `Shape_element` (see 4.2.810) in relation to other `Shape_element` that are not `Datum_system` (see 4.2.236).

NOTE EXPRESS specification:

```
*)
ENTITY position_tolerance_without_referenced_datum_system
    SUBTYPE OF (position_tolerance,
        geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- position_tolerance_without_referenced_datum_system
```

(*

4.2.698 Positional_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Positional_boundary` is a type of `Non_feature_shape_element` (see 4.2.560) that is a theoretical boundary that is defined by a `Boundary_based_position_tolerance` (see 4.2.77). The theoretical boundary is a boundary that cannot be intruded upon by the toleranced `Physical_feature_or_part_template` (see 4.2.658) if the `Physical_feature_or_part_template` is an internal `Physical_feature_or_part_template` (e.g., a hole) and the `Boundary_based_position_tolerance` is applied on a "maximum material principle" basis, or if the toleranced `Physical_feature_or_part_template` is an external `Physical_feature_or_part_template` (e.g., a tab) and the `Boundary_based_position_tolerance` is applied on a "least material requirement" basis. Additionally, the theoretical boundary is a boundary that cannot be escaped by the toleranced `Physical_feature_or_part_template` if the `Physical_feature_or_part_template` is an internal `Physical_feature_or_part_template` and the `Boundary_based_position_tolerance` is applied on a "least material requirement" basis, or if the toleranced `Physical_feature_or_part_template` is an external `Physical_feature_or_part_template` and the `Boundary_based_position_tolerance` is applied on a "maximum material principle" basis. A `Positional_boundary` is either a `Profile_related_positional_boundary` (see 4.2.750) or a `Dimension_related_positional_boundary` (see 4.2.295).

The data associated with a `Positional_boundary` are the following:

— `boundary_definition`.

4.2.698.1 `boundary_definition`

The `boundary_definition` is an inverse relationship that specifies that the existence of the `Positional_boundary` is dependent on the existence of the `Positional_boundary_definition` (see 4.2.700) that specifies the `Positional_boundary` as its `defined_positional_boundary`.

NOTE EXPRESS specification:

```

*)
ENTITY positional_boundary
  ABSTRACT SUPERTYPE OF (ONEOF (profile_related_positional_boundary,
    dimension_related_positional_boundary))
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    boundary_definition : positional_boundary_definition FOR
                          defined_positional_boundary;
END_ENTITY; -- positional_boundary

( *
```

4.2.699 `Positional_boundary_and_profile_boundary_member_relationship`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Positional_boundary_and_profile_boundary_member_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Positional_boundary_and_profile_boundary_member_relationship` is a relationship that associates a `Profile_related_positional_boundary` (see 4.2.750) with the `Tolerance_zone_boundary_member` (see 4.2.917) from which the `Profile_related_positional_boundary` is offset.

The data associated with a `Positional_boundary_and_profile_boundary_member_relationship` are the following:

- `base_profile_tolerance_zone_boundary_member`;
- `offset`;
- `offset_positional_boundary`.

4.2.699.1 `base_profile_tolerance_zone_boundary_member`

The `base_profile_tolerance_zone_boundary_member` specifies the `Tolerance_zone_boundary_member` (see 4.2.917) from which the `Profile_related_positional_boundary` (see 4.2.750) specified as the `offset_positional_boundary` is offset.

4.2.699.2 `offset`

The `offset` is an inverse relationship that specifies that the existence of the `Positional_boundary_and_profile_boundary_member_relationship` is dependent on the existence of the `Positional_boundary_offset` (see 4.2.703) that specifies the `Positional_boundary_and_profile_boundary_member_relationship` as its `offset_boundary_relationship`.

4.2.699.3 `offset_positional_boundary`

The `offset_positional_boundary` specifies the `Profile_related_positional_boundary` (see 4.2.750) that is offset from the `Tolerance_zone_boundary_member` (see 4.2.917) that is specified as the `base_profile_tolerance_zone_boundary_member`.

NOTE EXPRESS specification:

```

*)
ENTITY positional_boundary_and_profile_boundary_member_relationship
  SUBTYPE OF (managed_design_object);
    offset_positional_boundary          : profile_related_positional_boundary;
    base_profile_tolerance_zone_boundary_member : tolerance_zone_boundary_member;
  INVERSE
    offset : positional_boundary_offset FOR offset_boundary_relationship;
END_ENTITY; -- positional_boundary_and_profile_boundary_member_relationship

( *
```

4.2.700 Positional_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Positional_boundary_definition is a type of Managed_design_object (see 4.2.524). A Positional_boundary_definition is a definition that characterizes a Positional_boundary (see 4.2.698). A Positional_boundary_definition is either a Profile_related_positional_boundary_definition (see 4.2.751) or a Dimension_related_positional_boundary_definition (see 4.2.296).

The data associated with a Positional_boundary_definition are the following:

— defined_positional_boundary.

4.2.700.1 defined_positional_boundary

The defined_positional_boundary specifies the Positional_boundary (see 4.2.698) that is characterized by the Positional_boundary_definition.

NOTE EXPRESS specification:

```
*)
ENTITY positional_boundary_definition
  ABSTRACT SUPERTYPE OF (ONEOF (
    profile_related_positional_boundary_definition,
    dimension_related_positional_boundary_definition))
  SUBTYPE OF (managed_design_object);
  defined_positional_boundary : positional_boundary;
END_ENTITY; -- positional_boundary_definition

( *
```

4.2.701 Positional_boundary_member

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Positional_boundary_member is a type of Non_feature_shape_element (see 4.2.560) that is a member of a Dimension_related_positional_boundary (see 4.2.295). A Positional_boundary_member is defined by a Boundary_based_position_tolerance (see 4.2.77) in conjunction with a Dimensional_size (see 4.2.301).

The data associated with a Positional_boundary_member are the following:

— boundary_member_definition;

— composed_positional_boundary.

4.2.701.1 boundary_member_definition

The boundary_member_definition is an inverse relationship that specifies that the existence of the Positional_boundary_member is dependent on the existence of the Positional_boundary_member_definition (see 4.2.702) that specifies the Positional_boundary_member as its defined_positional_boundary_member.

4.2.701.2 composed_positional_boundary

The composed_positional_boundary is an inverse relationship that specifies that the existence of the Positional_boundary_member is dependent on the existence of the Dimension_related_positional_boundary (see 4.2.295) that specifies the Positional_boundary_member as one of its boundary_members.

NOTE EXPRESS specification:

```
*)
ENTITY positional_boundary_member
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    composed_positional_boundary : dimension_related_positional_boundary FOR
                                   boundary_members;
    boundary_member_definition   : positional_boundary_member_definition FOR
                                   defined_positional_boundary_member;
END_ENTITY; -- positional_boundary_member

(*
```

4.2.702 Positional_boundary_member_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Positional_boundary_member_definition is a type of Managed_design_object (see 4.2.524). A Positional_boundary_member_definition is a definition that characterizes a Positional_boundary_member (see 4.2.701). Each Positional_boundary_member_definition is defined by a Boundary_based_position_tolerance (see 4.2.77) in conjunction with a Dimensional_size (see 4.2.301). The shape of the Positional_boundary_member is identical to that of the toleranced Physical_feature_or_part_template (see 4.2.658). However, the size of the Positional_boundary_member is based on the associated Dimensional_size, the material conditions specified by the associated Boundary_based_position_tolerance, and whether the toleranced Physical_feature_or_part_template is an internal (e.g., a pair of opposing slot sides) or an external (e.g., a pair of opposing tab sides) Physical_feature_or_part_template.

The data associated with a `Positional_boundary_member_definition` are the following:

- `associated_dimension`;
- `boundary_definition`;
- `defined_positional_boundary_member`;
- `tolerance_value`.

4.2.702.1 associated_dimension

The `associated_dimension` specifies the `Dimensional_size` (see 4.2.301) that defines the maximum material size and minimum material size of the tolerated `Physical_feature_or_part_template` (see 4.2.658). If the associated `Boundary_based_position_tolerance` (see 4.2.77) is specified on a "maximum material principle" basis, the maximum material size is used as the base size to which the `Length_data_element` (see 4.2.511) specified as the `tolerance_value` is subtracted if the tolerated `Physical_feature_or_part_template` is an internal `Physical_feature_or_part_template`, or added if the tolerated `Physical_feature_or_part_template` is an external `Physical_feature_or_part_template`. If the associated `Boundary_based_position_tolerance` is specified on a "least material requirement" basis, the least material size is used as the base size to which the `Length_data_element` specified as the `tolerance_value` is added if the tolerated `Physical_feature_or_part_template` is an internal `Physical_feature_or_part_template`, or subtracted if the tolerated `Physical_feature_or_part_template` is an external `Physical_feature_or_part_template`.

4.2.702.2 boundary_definition

The `boundary_definition` is an inverse relationship that specifies that the existence of the `Positional_boundary_member_definition` is dependent on the existence of the `Dimension_related_positional_boundary_definition` (see 4.2.296) that specifies the `Positional_boundary_member_definition` as one of its `characterized_boundary_member_definitions`.

4.2.702.3 defined_positional_boundary_member

The `defined_positional_boundary_member` specifies the `Positional_boundary_member` (see 4.2.701) that is characterized by the `Positional_boundary_member_definition`.

4.2.702.4 tolerance_value

The `tolerance_value` specifies the `Length_data_element` (see 4.2.511) that represents the amount the size of the `Positional_boundary_member` (see 4.2.701) is reduced or enlarged from the base size that is determined by the associated `Dimensional_size` (see 4.2.301) and the material condition specified by the associated `Boundary_based_position_tolerance` (see 4.2.77).

Constraints:

WR1: The Dimension_related_positional_boundary specified as the composed_positional_boundary by the Positional_boundary_member that is specified as the defined_positional_boundary_member shall be the same instance of Dimension_related_positional_boundary as specified as the defined_positional_boundary by the Dimension_related_positional_boundary_definition that is specified as the boundary_definition.

WR2: The Positional_boundary_member_definition shall be specified by exactly one Geometric_tolerance as the tolerance_zone_or_boundary_definition, and that Geometric_tolerance shall be a Boundary_based_position_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY positional_boundary_member_definition
  SUBTYPE OF (managed_design_object);
    defined_positional_boundary_member : positional_boundary_member;
    tolerance_value                     : length_data_element;
    associated_dimension                : dimensional_size;
  INVERSE
    boundary_definition : dimension_related_positional_boundary_definition FOR
                           characterized_boundary_member_definitions;
END_ENTITY; -- positional_boundary_member_definition

( *
```

4.2.703 Positional_boundary_offset

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Positional_boundary_offset is a type of Managed_design_object (see 4.2.524). A Positional_boundary_offset is a definition that characterizes the offset of a Profile_related_positional_boundary (see 4.2.750) from a Tolerance_zone_boundary_member (see 4.2.917).

The data associated with a Positional_boundary_offset are the following:

- boundary_definition;
- offset_boundary_relationship;
- tolerance_value.

4.2.703.1 boundary_definition

The boundary_definition is an inverse relationship that specifies that the existence of the Positional_boundary_offset is dependent on the existence of the Profile_related_positional_boundary_definition (see 4.2.751) that specifies the Positional_boundary_offset as its boundary_offset.

4.2.703.2 offset_boundary_relationship

The offset_boundary_relationship specifies the Positional_boundary_and_profile_boundary_member_relationship (see 4.2.699) that associates the Profile_related_positional_boundary (see 4.2.750) with the Tolerance_zone_boundary_member (see 4.2.917) from which it is offset, and which is characterized by the Positional_boundary_offset.

4.2.703.3 tolerance_value

The tolerance_value specifies the Length_data_element (see 4.2.511) for the Positional_boundary_offset. The associated Profile_related_positional_boundary (see 4.2.750) is offset from the associated Tolerance_zone_boundary_member (see 4.2.917) by half the distance specified by the tolerance_value.

NOTE EXPRESS specification:

```

*)
ENTITY positional_boundary_offset
  SUBTYPE OF (managed_design_object);
    offset_boundary_relationship :
positional_boundary_and_profile_boundary_member_relationship;
    tolerance_value                : length_data_element;
  INVERSE
    boundary_definition : profile_related_positional_boundary_definition FOR
                           boundary_offset;
END_ENTITY; -- positional_boundary_offset

( *
  tt tc1 wd11 moved from entity to type
( *

```

4.2.704 Power_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Power_data_element is a

The data associated with a Power_data_element are the following:

— additional_data.

4.2.704.1 additional_dataConstraints:**UR1:****WR1:** NOTE EXPRESS specification:

```

*)
ENTITY power_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- power_data_element

```

```

( *

```

4.2.705 Pre_defined_datum_axis_3d_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_datum_axis_3d_symbol is a type of Non_feature_shape_definition (see 4.2.559) and a type of Bound_volume_shape (see 4.2.76). A Pre_defined_datum_axis_3d_symbol is a standard representation for a Datum_axis (see 4.2.215) in a 3 dimensional coordinate system.

The data associated with a Pre_defined_datum_axis_3d_symbol are the following:

— associated_element.

4.2.705.1 associated_element

The associated_element specifies the role of the Datum_axis (see 4.2.215) for the Pre_defined_datum_axis_3d_symbol.

NOTE EXPRESS specification:

```

*)
ENTITY pre_defined_datum_axis_3d_symbol
  SUBTYPE OF (bound_volume_shape, non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
WHERE
  WR1: NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
  WR2 : SELF\shape_definition.geometric_context.dimensionality = three_dimensional;
END_ENTITY; -- pre_defined_datum_axis_3d_symbol
( *

```

4.2.706 Pre_defined_datum_plane_3d_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_datum_plane_3d_symbol is a type of Non_feature_shape_definition (see 4.2.559) and a type of Bound_volume_shape (see 4.2.76). A Pre_defined_datum_plane_3d_symbol is a standard representation for a Datum_plane (see 4.2.232) in a 3 dimensional coordinate system. A Pre_defined_datum_plane_3d_symbol shall be parallel with the normal vector of the plane, and shall intersect the plane at the origin of the symbol.

The data associated with a Pre_defined_datum_plane_3d_symbol are the following:

— associated_element.

4.2.706.1 associated_element

The associated_element specifies the role of the Datum_plane (see 4.2.232) for the Pre_defined_datum_plane_3d_symbol.

NOTE EXPRESS specification:

```

*)
ENTITY pre_defined_datum_plane_3d_symbol
  SUBTYPE OF (bound_volume_shape, non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_plane;
WHERE
  WR1: NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
  WR3 : SELF\shape_definition.geometric_context.dimensionality = three_dimensional;
END_ENTITY; -- pre_defined_datum_plane_3d_symbol
( *
```

4.2.707 Pre_defined_dimension_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_dimension_qualifier is a type of Dimension_qualifier (see 4.2.294) the intent of which is unambiguous and is commonly used in geometric dimensioning.

The data associated with a Pre_defined_dimension_qualifier are the following:

— description.

4.2.707.1 description

The description specifies the label that describes the manner in which the Dimensional_characteristic (see 4.2.298) that are specified as the qualified_dimensions are qualified by the Pre_defined_dimension_qualifier. The description shall be either "average", or "free_state".

NOTE The attribute qualified_dimensions is declared in the Dimension_qualifier entity of which this entity is a subtype.

NOTE EXPRESS specification:

```
*)
ENTITY pre_defined_dimension_qualifier
  SUBTYPE OF (dimension_qualifier);
  description : dimension_qualifiers;
END_ENTITY; -- pre_defined_dimension_qualifier

( *
```

4.2.708 Pre_defined_geometric_tolerance_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_geometric_tolerance_qualifier is a type of Geometric_tolerance_qualifier (see 4.2.420) the intent of which is unambiguous and is commonly used in geometric tolerancing.

The data associated with a Pre_defined_geometric_tolerance_qualifier are the following:

— description.

4.2.708.1 description

The description specifies the label that describes the manner in which the Geometric_tolerance (see 4.2.419) that are specified as the qualified_geometric_tolerances are qualified by the Pre_defined_geometric_tolerance_qualifier. The description shall be either "each_element", "each_radial_element", "free_state", "major_diameter", "minor_diameter", "not_concave", "not_convex", or "pitch_diameter".

NOTE The attribute qualified_geometric_tolerances is declared in the Geometric_tolerance_qualifier entity of which this entity is a subtype.

EXAMPLE 1 "each element" is a commonly used qualifier that indicates the requirement specified by the geometric tolerance to which the qualifier is applied. "each_element" is on each line element of the feature to which the geometric tolerance is applied rather than on the feature as a whole.

NOTE EXPRESS specification:

```
*)
ENTITY pre_defined_geometric_tolerance_qualifier
  SUBTYPE OF (geometric_tolerance_qualifier);
  description : geometric_tolerance_qualifiers;
END_ENTITY; -- pre_defined_geometric_tolerance_qualifier

( *
```

4.2.709 Pre_defined_parallel_datum_axis_2d_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_parallel_datum_axis_2d_symbol is a type of Non_feature_shape_definition (see 4.2.559) and a type of Planar_shape (see 4.2.678). A Pre_defined_parallel_datum_axis_2d_symbol is a standard representation for a Datum_axis (see 4.2.215) in a 2 dimensional coordinate system. The axis represented shall be parallel to, or congruent with, the plane that represents the 2 dimensional coordinate system in an associated 3 dimensional coordinate system.

The data associated with a Pre_defined_parallel_datum_axis_2d_symbol are the following:

— associated_element.

4.2.709.1 associated_element

The associated_element specifies the role of the Datum_axis (see 4.2.215) for the Pre_defined_parallel_datum_axis_2d_symbol.

NOTE EXPRESS specification:

```
*)
ENTITY pre_defined_parallel_datum_axis_2d_symbol
  SUBTYPE OF (planar_shape, non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
WHERE
  WR1: NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
  WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_parallel_datum_axis_2d_symbol

( *
```

4.2.710 Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship` is a type of `Volume_shape_intersection` (see 4.2.960). A `Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship` establishes the geometric reference relationship between a `Datum_axis` (see 4.2.215) representation in a 3 dimensional coordinate system and the related representation in a 2 dimensional coordinate system where the `Datum_axis` is parallel to the plane that represents the 2 dimensional coordinate system in the 3 dimensional coordinate system.

NOTE Only the line representing the central axis is expected to be mathematically derived. Due to differences in 2 dimensional CAD systems the start and end treatments (arrowheads) may not be mathematically derived.

NOTE EXPRESS specification:

*)

```
ENTITY pre_defined_parallel_datum_axis_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_parallel_datum_axis_3d_symbol;
  SELF\volume_shape_intersection.projected_volume : pre_defined_datum_axis_3d_symbol;
END_ENTITY; -- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship
```

(*

4.2.711 Pre_defined_perpendicular_datum_axis_2d_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Pre_defined_perpendicular_datum_axis_2d_symbol` is a type of `Non_feature_shape_definition` (see 4.2.559) and a type of `Planar_shape` (see 4.2.678). A `Pre_defined_perpendicular_datum_axis_2d_symbol` is a standard representation for a `Datum_axis` (see 4.2.215) that is parallel to the normal of the plane that represents the 2 dimensional coordinated system in an associated 3 dimensional coordinate system.

The data associated with a `Pre_defined_perpendicular_datum_axis_2d_symbol` are the following:

— `associated_element`.

4.2.711.1 associated_element

The `associated_element` specifies the role of the `Datum_axis` (see 4.2.215) for the `Pre_defined_perpendicular_datum_axis_2d_symbol`.

NOTE EXPRESS specification:

```

*)
ENTITY pre_defined_perpendicular_datum_axis_2d_symbol
  SUBTYPE OF (planar_shape, non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
WHERE
  WR1: NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
  WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_perpendicular_datum_axis_2d_symbol
( *

```

4.2.712 Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship is a type of Volume_shape_intersection (see 4.2.960). A Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship establishes the geometric reference relationship between a Datum_axis (see 4.2.215) representation in a 3 dimensional coordinate system and the related representation in a 2 dimensional coordinate system where the Datum_axis is perpendicular to the plane that represents the 2 dimensional coordinate system in the 3 dimensional coordinate system. The controlled information is the centre of the circle.

NOTE Only the center of the circle representing the central axis is expected to be directly referenced. Due to differences in CAD systems resolutions the lines used as crosshairs may slightly deviate from intersecting at the center of the circle.

NOTE EXPRESS specification:

```

*)
ENTITY pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_perpendicular_d
  SELF\volume_shape_intersection.projected_volume : pre_defined_datum_axis_3d_symbol;
END_ENTITY; -- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship
( *

```

4.2.713 Pre_defined_perpendicular_datum_plane_2d_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pre_defined_perpendicular_datum_plane_2d_symbol is a type of Non_feature_shape_definition (see 4.2.559) and a type of Planar_shape (see 4.2.678). A Pre_defined_perpendicular_plane_2d_symbol is a standard representation for a Datum_plane (see 4.2.232) that is perpendicular to the plane that represents the 2 dimensional coordinate system in an associated 3 dimensional coordinate system.

The data associated with a `Pre_defined_perpendicular_datum_plane_2d_symbol` are the following:

— `associated_element`.

4.2.713.1 `associated_element`

The `associated_element` specifies the role of the `Datum_plane` (see 4.2.232) for the `Pre_defined_perpendicular_datum_plane_2d_symbol`.

NOTE EXPRESS specification:

```
*)
ENTITY pre_defined_perpendicular_datum_plane_2d_symbol
  SUBTYPE OF (planar_shape, non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
WHERE
  WR1: NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
  WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_perpendicular_datum_plane_2d_symbol
(*
```

4.2.714 `Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship` is a type of `Volume_shape_intersection` (see 4.2.960). A `Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship` establishes the geometric reference relationship between a `Datum_plane` (see 4.2.232) representation in a 3 dimensional coordinate system and the related representation in a 2 dimensional coordinate system where the `Datum_plane` is perpendicular to the plane that represents the 2 dimensional coordinate system in the 3 dimensional coordinate system. The controlled information is the line representing the normal of the plane.

NOTE Due to differences in CAD systems resolutions the lines used as base and arrowhead may slightly deviate from intersecting at the start and end of the line.

NOTE EXPRESS specification:

```
*)
ENTITY pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_perpendicular_d
  SELF\volume_shape_intersection.projected_volume : pre_defined_datum_plane_3d_symbol;
END_ENTITY; -- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship
(*
```

4.2.715 Precision_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Precision_qualifier is a type of Managed_design_object (see 4.2.524) in which there is an associated precision. The precision is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY precision_qualifier
  SUBTYPE OF (managed_design_object);
    precision_value : integer;
END_ENTITY; -- precision_qualifier
(*
```

4.2.716 Pressure_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Pressure_data_element is a

The data associated with a Pressure_data_element are the following:

— additional_data.

4.2.716.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY pressure_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- pressure_data_element
(*
```

4.2.717 Primary_orientation_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Primary_orientation_feature is a type of Part_feature (see 4.2.609) and is a type of Datum_feature (see 4.2.226). A Primary_orientation_feature participates in providing for the unambiguous orientation of a part on its next assembly.

NOTE A Primary_orientation_feature is not necessarily also a Primary_reference_terminal (see 4.2.718).

EXAMPLE 1 Figure 85 illustrates two of the three-dimensional geometric relationships between a component and the printed circuit board it is mounted on. A Datum_system (see 4.2.236) is included for reference purposes. The Z direction is parallel to the normal of the top surface of the board. Both the top and bottom surface normal vectors are parallel to the normal vectors of the seating plane for the components.

EXAMPLE 2 Figure 86 illustrates two of the two-dimensional geometric relationships between a component and the printed circuit board it is mounted on. A Datum_system (see 4.2.236) is included for reference purposes.

EXAMPLE 3 Figures 87 through 91 illustrates orientation details for microcircuit packages.

EXAMPLE 4 Figure 92 illustrates details associated with several variations of an Axial package modification and the associated seating_plane. In all these cases, the seating plane orientation with respect to the Axis_placement (see 4.2.56) is constant, while the Package (see 4.2.573) shape orientation changes due to the modification.

EXAMPLE 5 Figure 93 illustrates Primary_reference_terminal (see 4.2.718) assignment and also assignment of Polarity_indication_feature. Knowledge of Polarity_indication_features separate from the orientation feature increases the ability to determine when to use polarity indication symbols on the interconnect substrate for polarized components e.g., polarized capacitors, diodes).

NOTE EXPRESS specification:

```
*)
ENTITY primary_orientation_feature
  SUBTYPE OF (part_feature, datum_feature);
    associated_body_vertical_extent : OPTIONAL SET [1:2] OF package_body_surface;
END_ENTITY; -- primary_orientation_feature

( *
```

4.2.718 Primary_reference_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

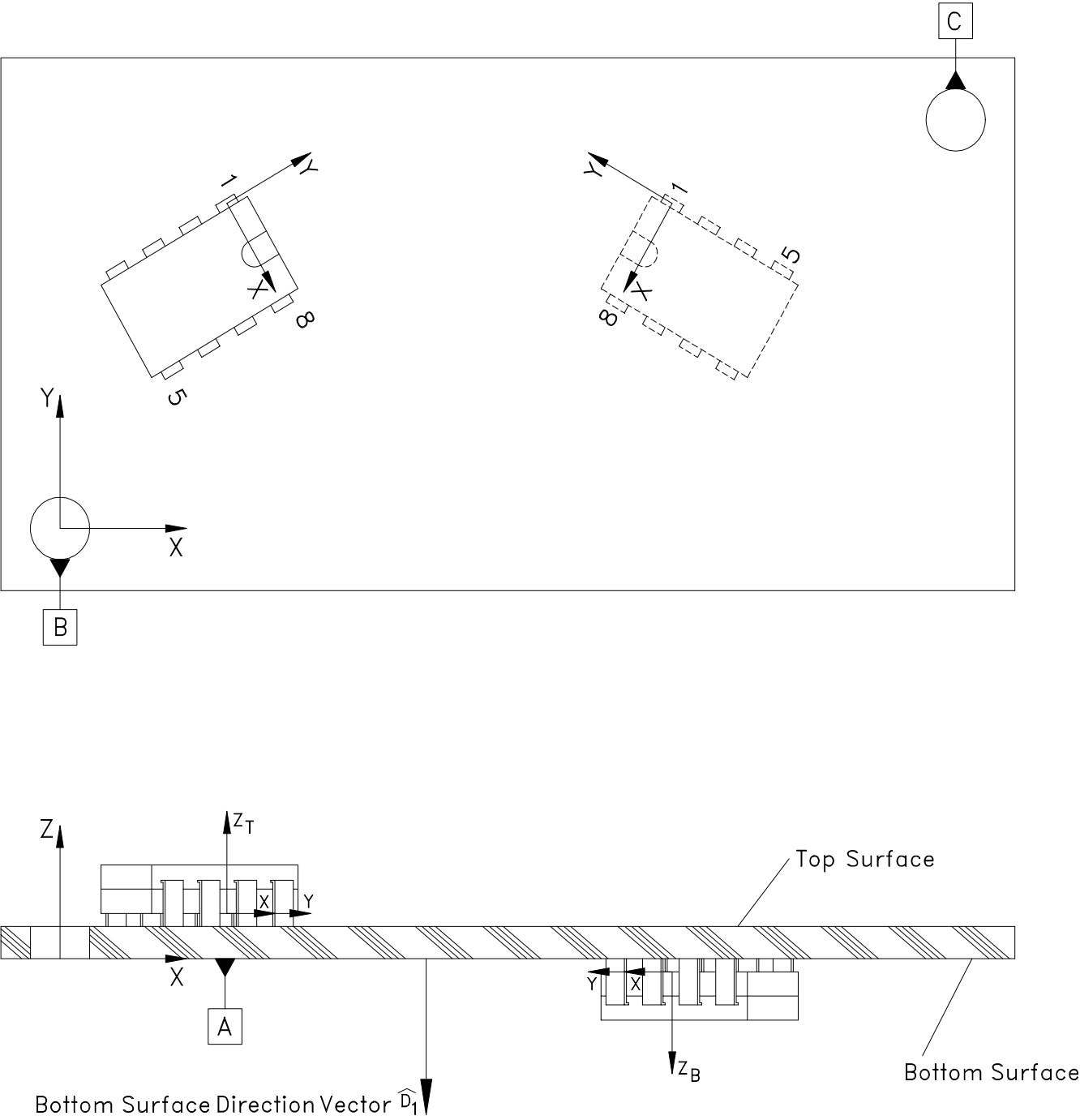


Figure 85 – Three-dimensional assembly relationships

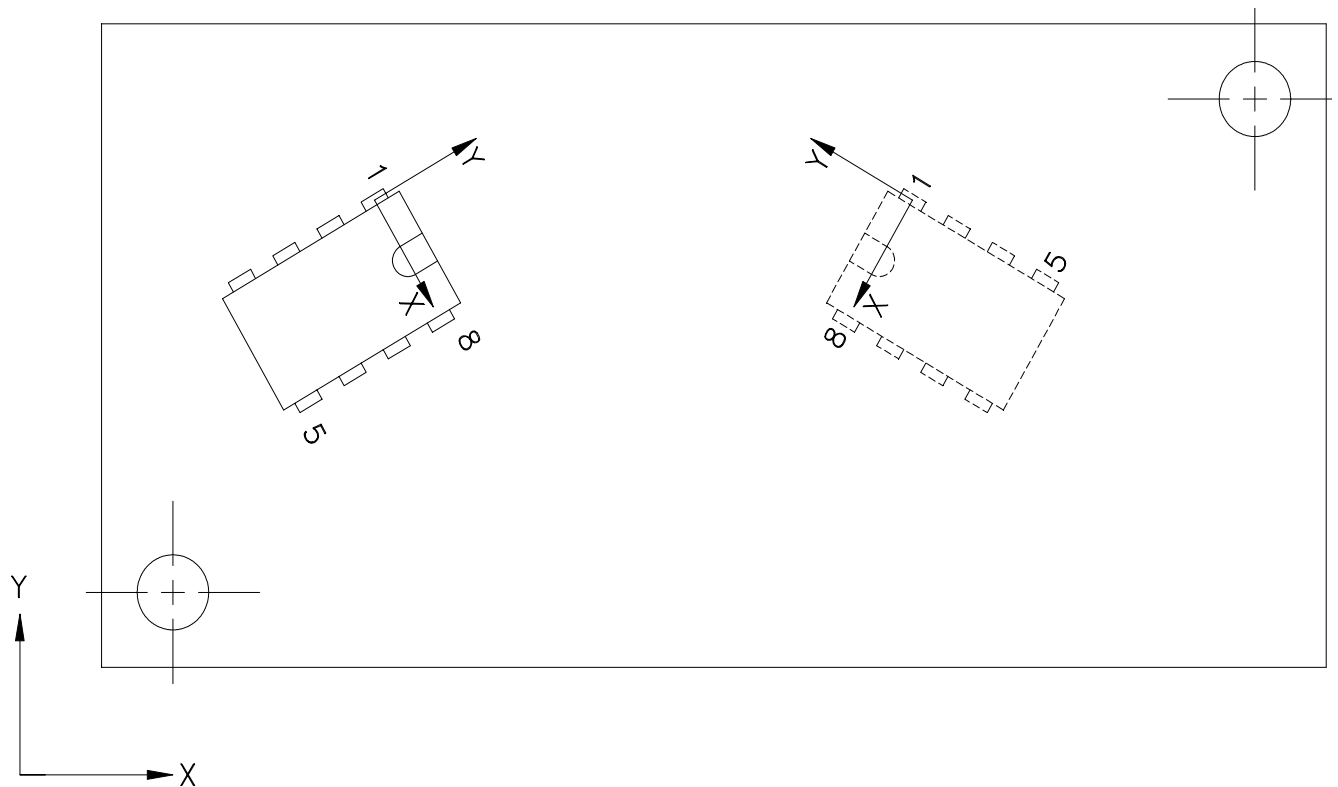


Figure 86 – Two-dimensional assembly relationships

A `Primary_reference_terminal` is a type of `Package_terminal` (see 4.2.581). This terminal is assigned the first identifier string in the sequence of package terminal names to be assigned to the part. The `Primary_reference_terminal` shall not specify a `precedent_feature` and is the first member of the sequence of `Package_terminal`. The sort order of the names assigned to the `Packaged_terminals` is defined by the explicit sequence established by the use of `precedent_feature` inherited from `Part_feature`.

EXAMPLE 1 If a value of '1' is available, assigning a value of '2' to this terminal would be prohibited by this part of ISO 10303, due to the possibility of confusion.

NOTE The `Primary_reference_terminal` is often associated with an orientation feature, but it may not be the orientation feature itself. Usually the `Primary_reference_terminal` and the `Primary_orientation_feature` (see 4.2.717) are different `part_features`.

Constraints:

WR1: The `precedent_feature` inherited from `Part_feature` shall not be provided for a `Primary_reference_terminal`.

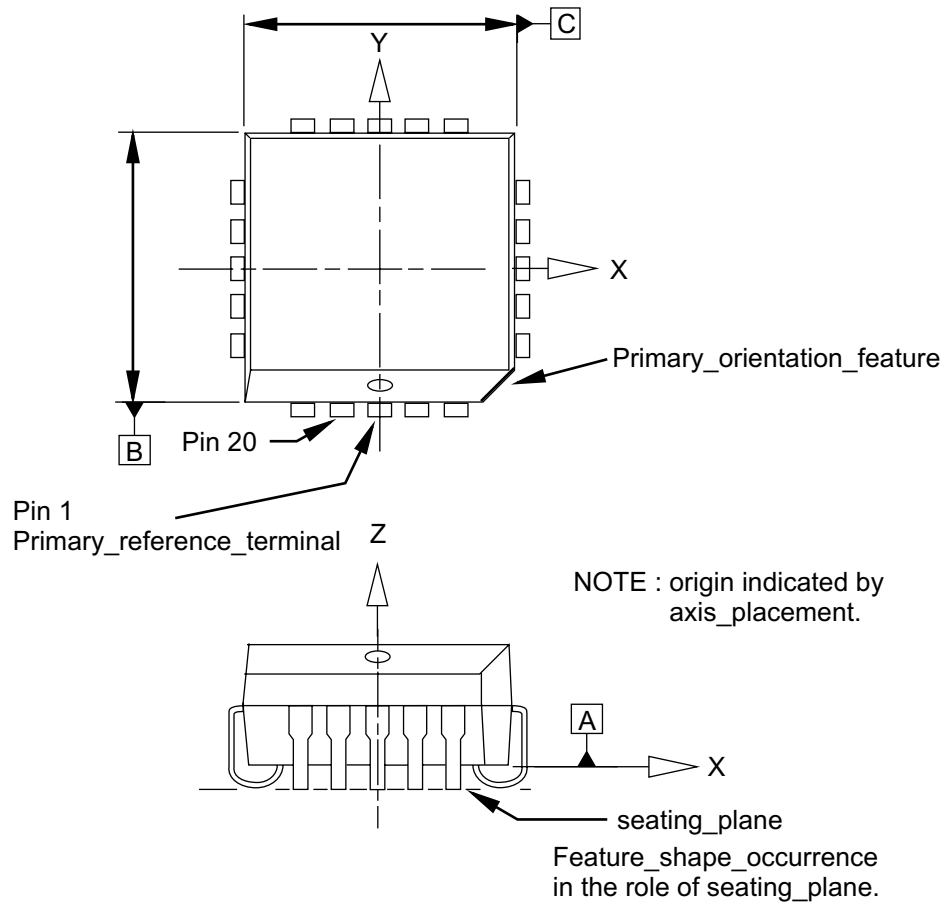


Figure 87 – Twenty pin orientation features

NOTE EXPRESS specification:

```

*)
ENTITY primary_reference_terminal
  SUBTYPE OF (package_terminal);
  --tt tc1 wd12 added rule
WHERE
  WR1: NOT(EXISTS(SELF\part_feature.precedent_feature));
END_ENTITY; -- primary_reference_terminal

( *
```

4.2.719 Primary_stratum_indicator_symbol

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

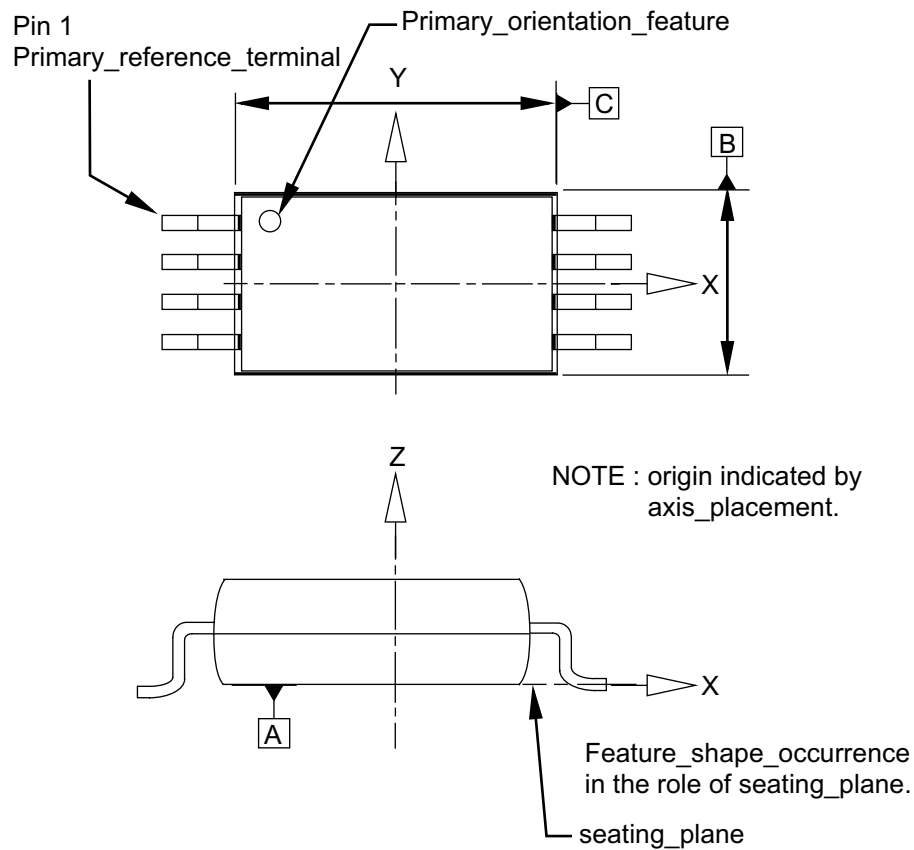


Figure 88 – Eight lead gull wing orientation features

A `Primary_stratum_indicator_symbol` is a type of `Special_symbol_laminate_component` (see 4.2.834) that is used to identify the arbitrarily named primary side of a printed circuit board.

NOTE This symbol may be found in the stratum material of the conductive stratum that is on the primary side of the printed circuit board, implemented in a stratum of silk-screen printed marking ink on the primary side, or otherwise. The primary side is normally defined as that side of the printed circuit board which will contain the largest number of components, but in cases where the component count is balanced, the primary side is arbitrarily named for convenience. Not all printed circuit boards contain this symbol.

Constraints:

WR1: The value of the attribute `primary_design_layer_stratum` shall be TRUE for the `Design_layer_stratum` in the role of `resident_stratum` for the member of `Stratum_feature` in the role of `stratum_feature_implementation` inherited from the supertype `Stratum_feature_template_component`.

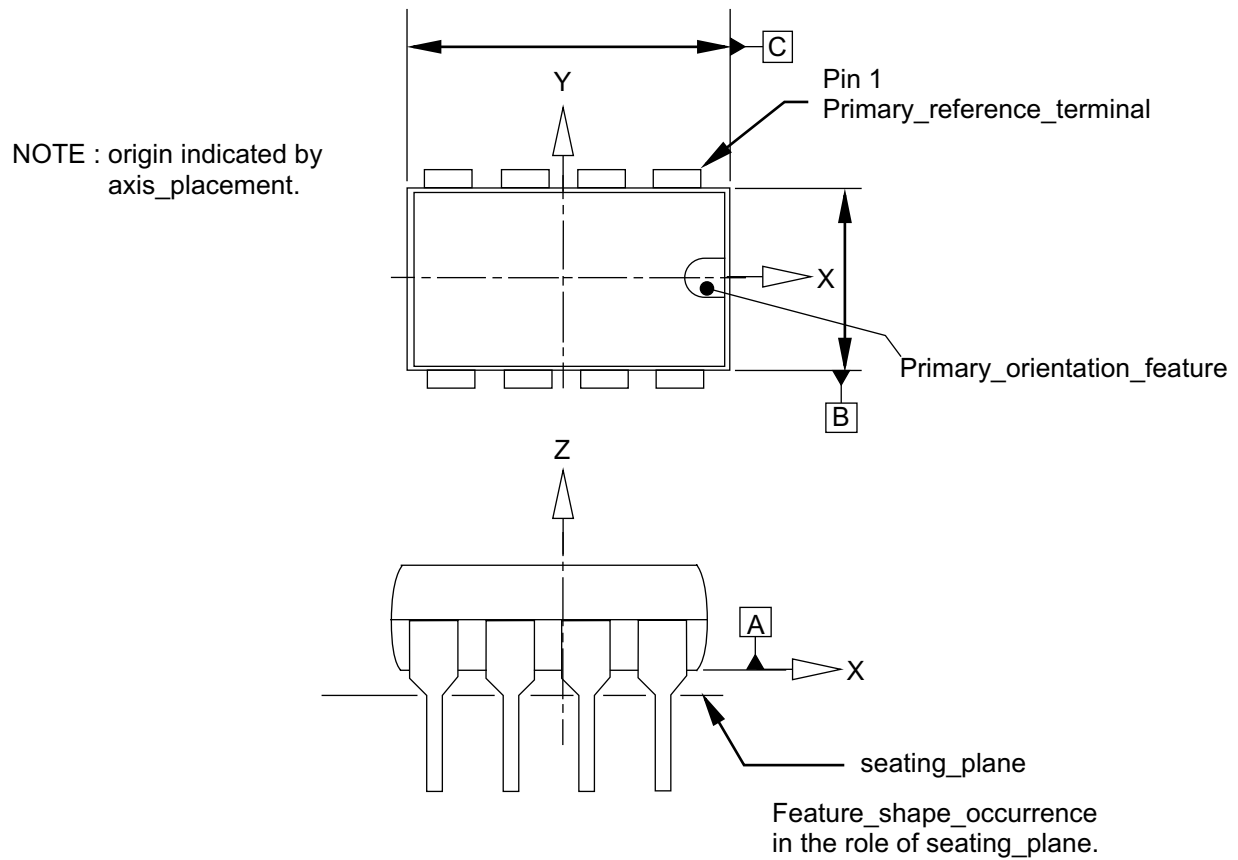


Figure 89 – Eight lead DIP orientation features

NOTE EXPRESS specification:

```

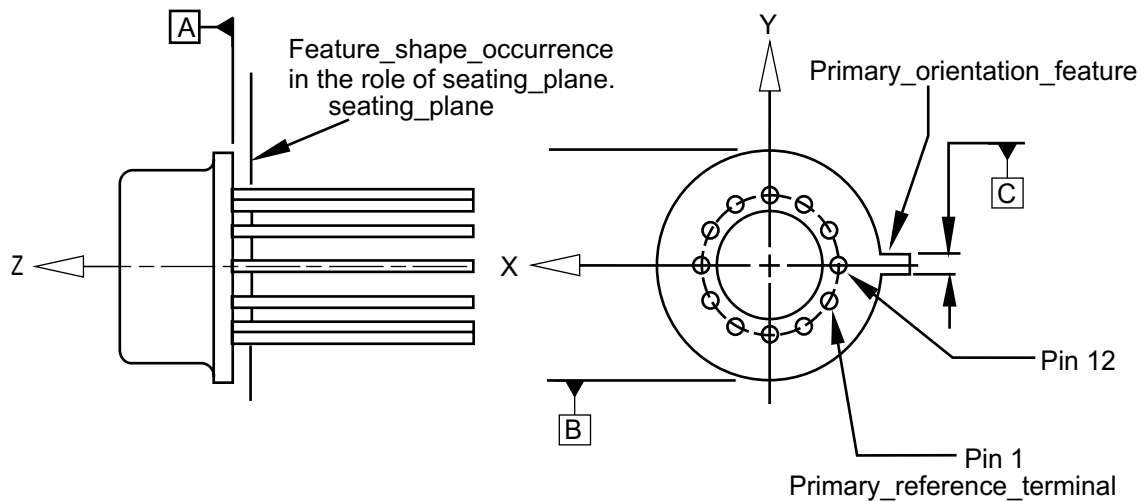
*)
ENTITY primary_stratum_indicator_symbol
  SUBTYPE OF (special_symbol_laminate_component);
END_ENTITY; -- primary_stratum_indicator_symbol

--tt tc1 correction
(

```

4.2.720 Primitive_2d

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.



NOTE : axis_placement at origin.

Figure 90 – TO-99 orientation features

A Primitive_2d is a type of Shape_item (see 4.2.818). A Primitive_2d may be one of a Circular_area (see 4.2.101), an Elliptic_area (see 4.2.350), a Rectangular_area (see 4.2.762), or a Polygonal_area (see 4.2.694).

Constraints:

WR1: The dimensionality attribute of the Cartesian_coordinate_system (see 4.2.86) referenced by the geometric_context attribute inherited from the Feature_shape_definition (see 4.2.372) shall be two-dimensional within a population of Primitive_2d.

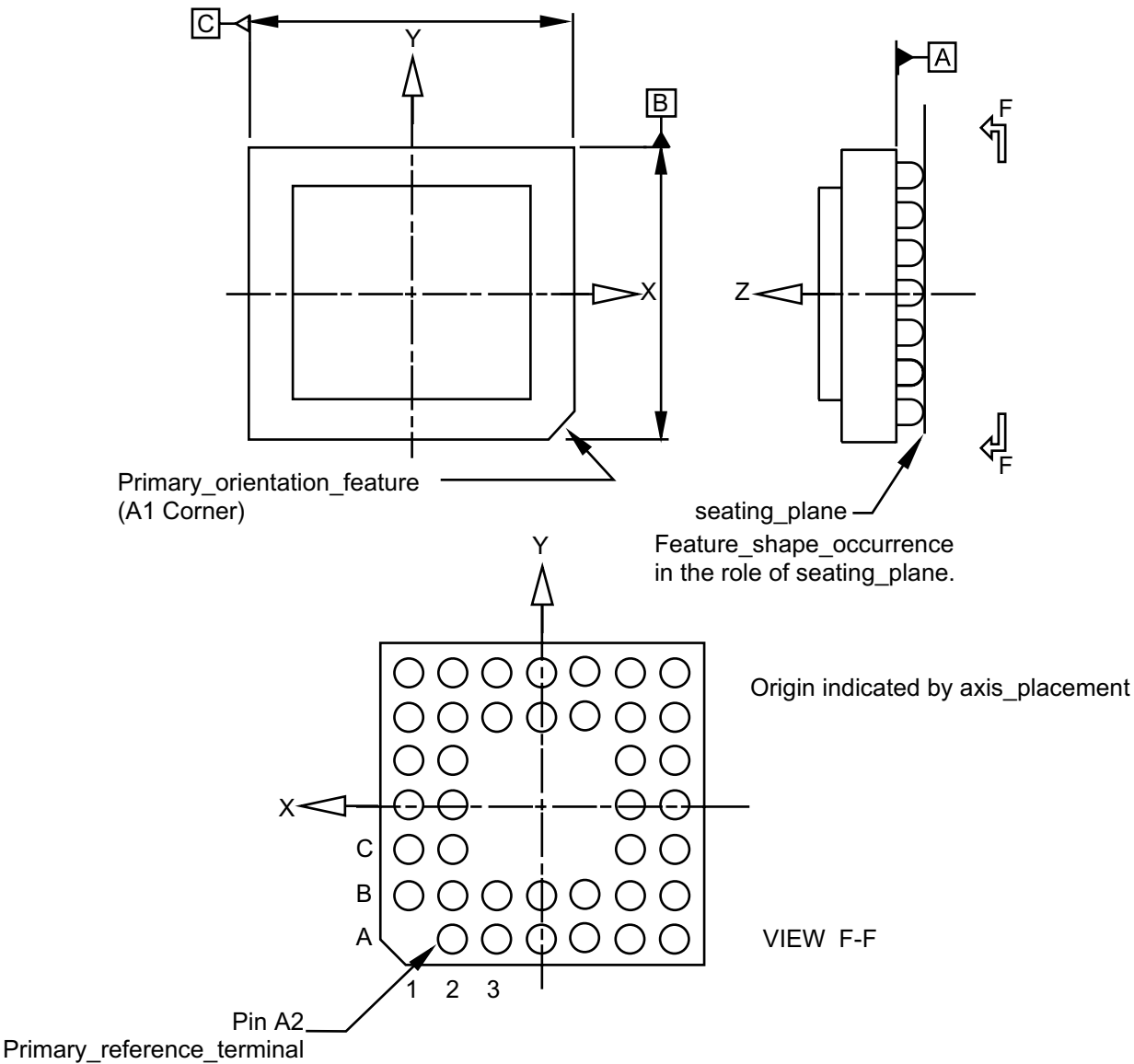


Figure 91 – Ball grid array orientation features

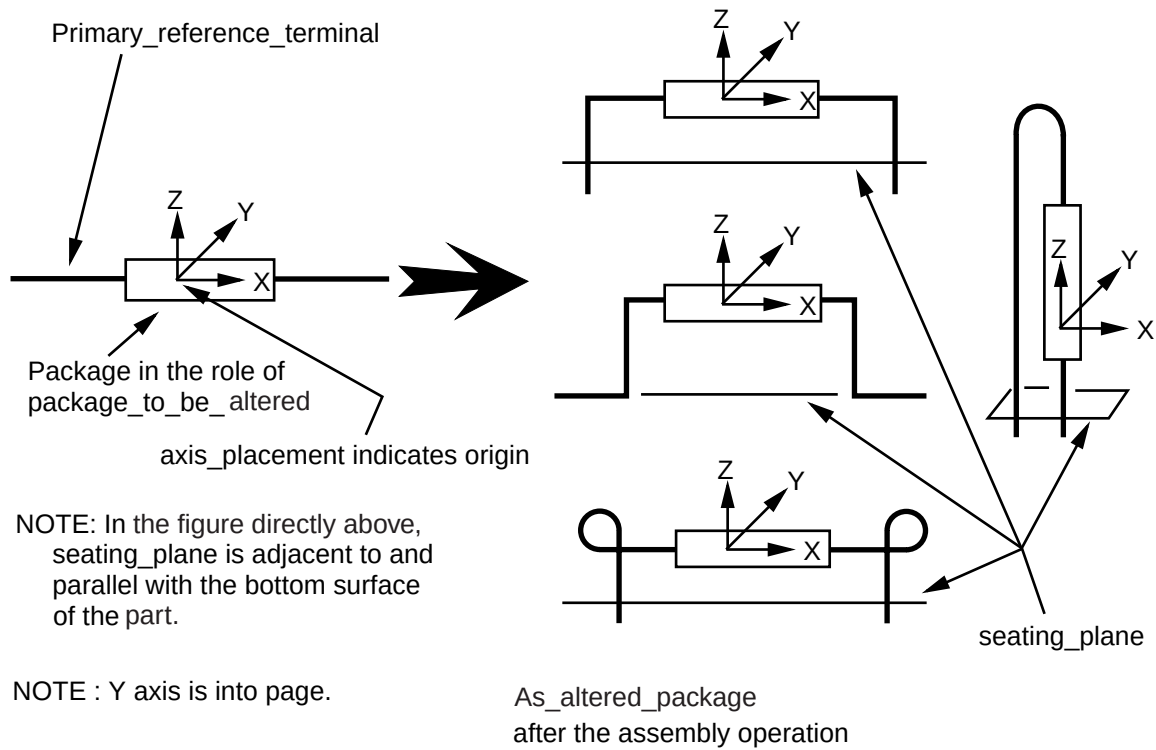


Figure 92 – Axial Package modifications and resulting orientations

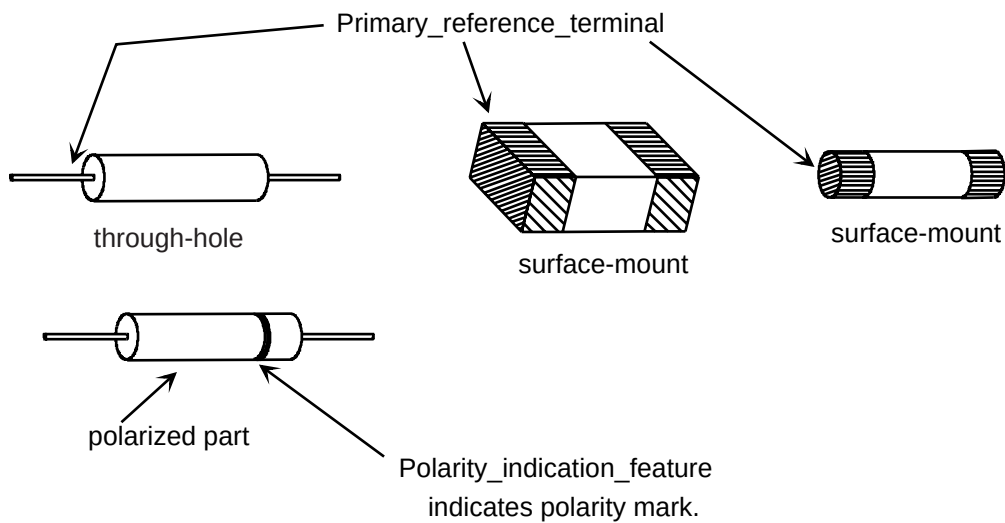


Figure 93 – Primary_reference_terminal and Polarity_indication_feature for discrete devices

NOTE EXPRESS specification:

```

*)
ENTITY primitive_2d
--tt tc9 modified
    ABSTRACT SUPERTYPE OF (ONEOF (circular_area,
                                   elliptic_area,
                                   rectangular_area,
                                   polygonal_area))
    SUBTYPE OF (shape_item);
WHERE
    WR1 : SELF\shape_item.geometric_context.dimensionality
                                                = two_dimensional;
END_ENTITY; -- rectangular_area

--tt tc1 wd8 modified
(

```

4.2.721 Printed_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_component is a type of Assembly_component (see 4.2.34). A Printed_component may be a Join_two_physical_connectivity_definition_supporting_printed_component (see 4.2.481) or a Printed_connector_component (see 4.2.724) or a Routed_printed_component (see 4.2.779). A Printed_component is an occurrence of Printed_part_template (see 4.2.730) that is used as an Assembly_component (see 4.2.34) in an Interconnect_module (see 4.2.440). A Printed_component is manufactured integral to an Interconnect_module (see 4.2.440). Only the Routed_printed_component subtype shall reference a Printed_part_cross_section_template as a usage_definition.

The data associated with a Printed_component are the following:

- access_mechanism;
- cad_layer;
- stratum_feature_implementation;
- usage_definition.

4.2.721.1 access_mechanism

The access_mechanism is an inverse relationship that specifies that the existence of the Printed_component is dependent on the existence of the Printed_component_join_terminal (see 4.2.722) that specifies the Printed_component as its associated_component. There shall be one or more Printed_component_join_terminal for a Printed_component.

4.2.721.2 cad_layer

The `cad_layer` specifies the `External_definition` (see 4.2.361) for the `Printed_component`. The `cad_layer` is optional and need not be supplied.

NOTE The `cad_layer` is only appropriate where there is a single layer material being defined.

4.2.721.3 stratum_feature_implementation

The `stratum_feature_implementation` specifies the `Stratum_feature` (see 4.2.851) for the `Printed_component`. There shall be one or more `Stratum_feature` for a `Printed_component`.

4.2.721.4 usage_definition

The `usage_definition` specifies the `Printed_part_template` (see 4.2.730) or the `Printed_part_cross_section_template` (see 4.2.728) for the definition for the `Printed_component`.

NOTE EXPRESS specification:

*)

ENTITY printed_component

SUPERTYPE OF (ONEOF(layout_macro_component,
printed_connector_component,
routed_printed_component))

SUBTYPE OF (assembly_component);

cad_layer : OPTIONAL external_definition;
stratum_feature_implementation : OPTIONAL SET [1:?] OF stratum_feature;
SELF\assembly_component.usage_definition :
printed_part_template_or_printed_part_cross_section_template;
printed_component_stack : OPTIONAL SET [1:?] OF printed_component_link;
required_material_stack : OPTIONAL layer;

DERIVE

access_mechanism : SET [1:?] OF printed_component_join_terminal :=
raw_access_mechanism +
printed_component_access_mechanisms(printed_component_stack);

INVERSE

raw_access_mechanism : SET [1:?] OF printed_component_join_terminal FOR
associated_component;
associated_precedent_link : SET [0:1] OF printed_component_link FOR
precedent_printed_component;
associated_subsequent_link : SET [0:1] OF printed_component_link FOR
subsequent_printed_component;

WHERE

WR1: (NOT EXISTS(stratum_feature_implementation) OR
same_stratum(stratum_feature_implementation)) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
--either a printed component is a single layer self contained or is defined by the
--links entirely.
--defining things depend on whether the right stuff is included in the definition
--if self is a layout_macro_component, then the rules are defined in that object.
WR2: (NOT ((EXISTS (SELF\assembly_component.usage_definition.of_stratum_technology)) XOR
EXISTS(stratum_feature_implementation))) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
WR3: (NOT ((EXISTS (SELF\assembly_component.usage_definition.material_stack)) XOR
EXISTS(required_material_stack))) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
WR4: (NOT ((EXISTS (SELF\assembly_component.usage_definition.printed_part_template_stack
EXISTS(printed_component_stack))) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
WR5: (NOT (EXISTS(printed_component_stack)) OR
consistent_stack_definition(SELF\assembly_component.usage_definition,
printed_component_stack)) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
WR6: (NOT (EXISTS(associated_precedent_link) OR EXISTS(associated_subsequent_link)) OR
EXISTS(stratum_feature_implementation)) OR
('AP210_ARM.LAYOUT_MACRO_COMPONENT' IN TYPEOF(SELF));
--if this is used by a link, there shall be no further decomposition.
WR7: NOT('AP210_ARM.ROUTED_PRINTED_COMPONENT' IN TYPEOF(SELF)) OR

```

        ( 'AP210_ARM.PRINTED_PART_CROSS_SECTION_TEMPLATE'
          IN TYPEOF(SELF\assembly_component.usage_definition));
WR8: NOT( 'AP210_ARM.PRINTED_PART_CROSS_SECTION_TEMPLATE'
          IN TYPEOF(SELF\assembly_component.usage_definition)) OR
        ( 'AP210_ARM.ROUTED_PRINTED_COMPONENT'
          IN TYPEOF(SELF));
END_ENTITY; -- printed_component
( *

```

4.2.722 Printed_component_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_component_join_terminal is a type of Component_terminal (see 4.2.137). This terminal is a terminus for internal circuitry in the interconnect substrate.

EXAMPLE 1 Figure 94 illustrates a detail of a layout drawing that shows the internal terminals of the printed connector (edge connector).

The data associated with a Printed_component_join_terminal are the following:

- associated_component;
- connection;
- definition;
- stratum_concept_implementation_component.

4.2.722.1 associated_component

The associated_component specifies the Printed_component (see 4.2.721) being accessed by the Printed_component_join_terminal.

4.2.722.2 connection

The connection specifies the role of the Connection_zone (see 4.2.185) for the Printed_component_join_terminal.

4.2.722.3 definition

The definition specifies the Printed_part_template_join_terminal (see 4.2.733) that defines the Printed_component_join_terminal.

4.2.722.4 stratum_concept_implementation_component

The stratum_concept_implementation_component specifies either the Inter_stratum_feature (see 4.2.432), the Laminate_component (see 4.2.483), the Stratum (see 4.2.845), the Stratum_feature (see 4.2.851), or the Stratum_surface (see 4.2.868) for the Printed_component_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY printed_component_join_terminal
  SUBTYPE OF (laminate_component_join_terminal);
  SELF\component_feature.definition      : printed_part_template_join_terminal;
  SELF\component_feature.associated_component : printed_component;
  stratum_concept_implementation        : stratum_feature;
WHERE
  WR1: SIZEOF(QUERY ( sf &#60;* SELF\component_feature.
    associated_component\printed_component.
      stratum_feature_implementation |
      sf = stratum_concept_implementation )
    ) = 1;
END_ENTITY; -- printed_component_join_terminal

--tt tc1 wd9 added
( *
```

4.2.723 Printed_component_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_component_link is a type of Managed_design_object (see 4.2.524).

NOTE EXPRESS specification:

```

*)
ENTITY printed_component_link
  SUBTYPE OF (managed_design_object);
    precedent_printed_component : printed_component;
    subsequent_printed_component : printed_component;
    definition : printed_part_template_link;
INVERSE
  associated_printed_component : printed_component FOR
    printed_component_stack;
WHERE
  WR1: precedent_printed_component\assembly_component.usage_definition =
    definition.precedent_printed_part_template;
  WR2: subsequent_printed_component\assembly_component.usage_definition =
    definition.subsequent_printed_part_template;
  WR3: precedent_printed_component &#60;&#62; subsequent_printed_component;
END_ENTITY; -- printed_component_link

( *
```

4.2.724 Printed_connector_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_connector_component is a type of Printed_component (see 4.2.721). The Printed_connector_component is the occurrence of a printed_connector_template. The purpose of the Printed_connector_component is to provide access to the assembly in which the interconnect substrate, that the realized connector is part of, is installed.

The data associated with a Printed_connector_component are the following:

- interface_access_mechanism;
- usage_definition.

4.2.724.1 interface_access_mechanism

The interface_access_mechanism is an inverse relationship that specifies that the existence of the Printed_connector_component is dependent on the existence of the Printed_connector_component_interface_terminal (see 4.2.725) that specifies the Printed_connector_component as its associated_component. There shall be one or more Printed_connector_component_interface_terminal for a Printed_connector_component.

4.2.724.2 usage_definition

The usage_definition specifies the role of the Printed_connector_template (see 4.2.726) for the Printed_connector_component.

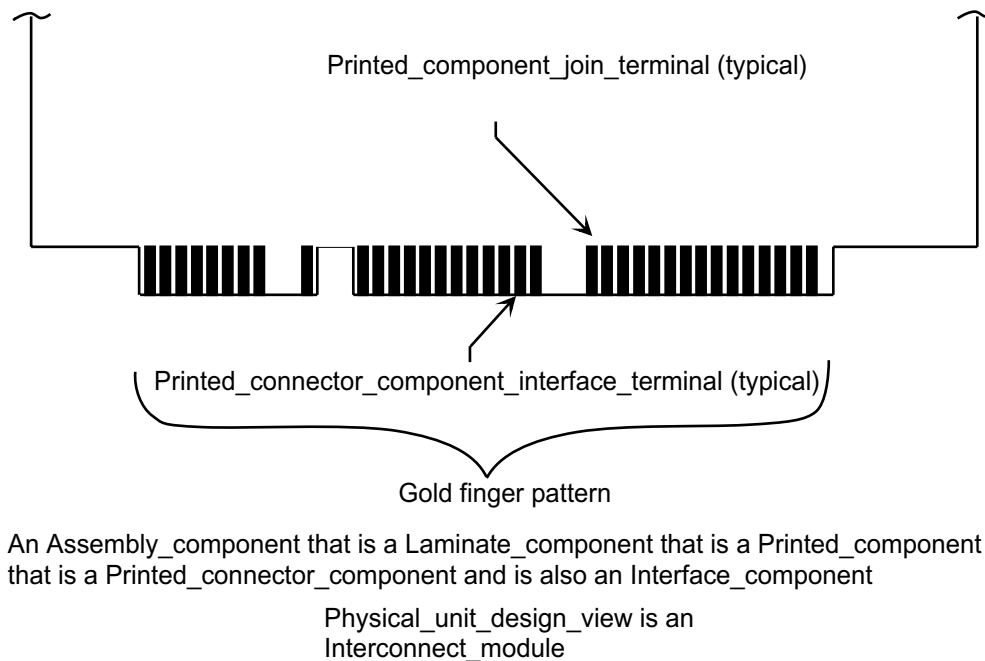


Figure 94 – Printed_connector_component and associated terminals

NOTE EXPRESS specification:

```

*)
ENTITY printed_connector_component
  SUBTYPE OF (printed_component);
  SELF\assembly_component.usage_definition : printed_connector_template;
  INVERSE
    interface_access_mechanism : SET [1:?] OF
      printed_connector_component_interface_terminal FOR
      associated_component;
END_ENTITY; -- printed_connector_component

--tt tc1 wd9 modified
--tt tc1 wd8 modified
(

```

4.2.725 Printed_connector_component_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_connector_component_interface_terminal is a type of Component_feature (see 4.2.118). The Printed_connector_component_interface_terminal is an individual access mechanism (e.g., one terminal on a Printed_connector_component). "Interface" means it is not tied to the substrate as part of the fabrication or assembly operation.

The data associated with a Printed_connector_component_interface_terminal are the following:

- associated_component;
- definition.

4.2.725.1 associated_component

The associated_component specifies the role of the Printed_connector_component (see 4.2.724) for the Printed_connector_component_interface_terminal.

4.2.725.2 definition

The definition specifies the role of the Printed_part_template_interface_terminal (see 4.2.732) for the Printed_connector_component_interface_terminal.

NOTE Use of this capability occurs when a member of Component_feature_to_physical_usage_view_assignment references a member of Printed_connector_component_interface_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY printed_connector_component_interface_terminal
  SUBTYPE OF (laminate_component_interface_terminal);
  SELF\component_feature.definition :
      printed_part_template_interface_terminal;
  SELF\component_feature.associated_component : printed_connector_component;
--tt tc1 wd9 modified
  stratum_concept_implementation : stratum_feature;
WHERE
  WR1: SIZEOF(QUERY( temp &#60;*
    SELF\component_feature.associated_component\printed_component.stratum_feature_implemen
    | temp :=: stratum_concept_implementation )) = 1;
END_ENTITY; -- printed_connector_component_interface_terminal

( *
```

4.2.726 Printed_connector_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_connector_template is a type of a Printed_part_template (see 4.2.730). A Printed_connector_template is a Printed_part_template that contains contacts, each of whose function is to act as a short from an interface_terminal of a Printed_part_template to the related join_terminal of the Printed_part_template. Each contact corresponds to one member of characterized_connector_terminal_relationship. The contact shall be a Part_template and the geometric relationship of the shape of the contact to the shape of the Printed_connector_template shall be established by a member of Usage_view_usage_relationship.

EXAMPLE 1 Figure 95 illustrates a template and associated parameters for one contact of a printed connector. Specific parameter values and explicit geometry will be provided for individual technology based contacts.

EXAMPLE 2 Figure 94 illustrates a detail of a layout drawing that shows an instance of the Printed_connector_template.

NOTE 1 It is redundant to have a separate functional definition that merely specifies shorts for any type of connector in this part of ISO 10303, because the functional definition of a short is included in the definition of the connector. That is, a functional definition that worries about shorts is not a functional definition. It is appropriate to define the location of required shorts in the physical definition, and connector is one of the places in this part of ISO 10303 where that occurs.

NOTE 2 There may be multiple simulation models associated with a connector in this part of ISO 10303. The explicit method of defining the connector functionality does not prohibit having multiple simulation models. This explicit method is for the purpose of supporting the standardized structural definition of connectivity.

The data associated with a Printed_connector_template are the following:

— characterized_connector_terminal_relationship.

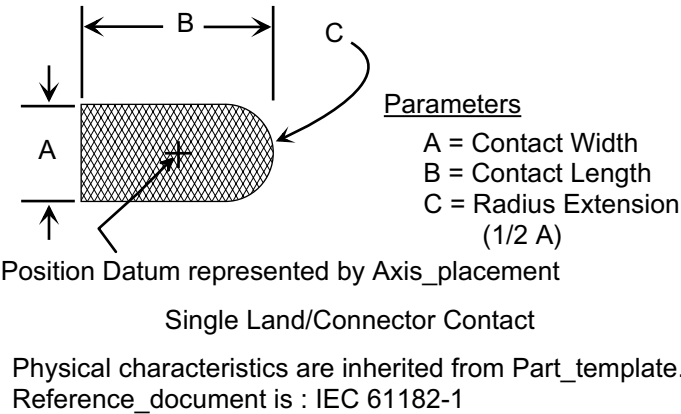


Figure 95 – Single contact of a Printed_connector_template

4.2.726.1 characterized_connector_terminal_relationship

The characterized_connector_terminal_relationship is an inverse relationship that specifies that the existence of the Printed_connector_template is dependent on the existence of the Printed_connector_template_terminal_relationship (see 4.2.727) that specifies the Printed_connector_template as its connector. There shall be one or more Printed_connector_template_terminal_relationship for a Printed_connector_template.

Constraints:

WR1: The material of individual contacts shall be disjoint.

WR2: A member of Usage_view_usage_relationship shall be populated for each member of characterized_connector_terminal_relationship to position the Part_template referenced indirectly by Printed_connector_template_terminal_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY printed_connector_template
  SUBTYPE OF (printed_part_template);
  INVERSE
    characterized_connector_terminal_relationship :
SET [1:?] OF printed_connector_template_terminal_relationship FOR connector;
END_ENTITY; -- printed_connector_template

( *
```

4.2.727 Printed_connector_template_terminal_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_connector_template_terminal_relationship is a type of Managed_design_object (see 4.2.524). A Printed_connector_template_terminal_relationship is the definition of the connection between a Printed_part_template_interface_terminal (see 4.2.732) and a Printed_part_template_join_terminal (see 4.2.733) for one contact of a Printed_connector_template.

NOTE 1 An occurrence of Printed_connector_template_terminal_relationship is only required at the Part model level in the Standard. A Printed_connector_component (see 4.2.724) that references a Printed_connector_template (see 4.2.726) has the visibility of this functional requirement by virtue of a mandatory inverse relationship to the Printed_connector_template_terminal_relationship on the Printed_connector_template.

NOTE 2 A Printed_connector_template_terminal_relationship is the actual specification of the short referred to in Printed_connector_template.

The data associated with a Printed_connector_template_terminal_relationship are the following:

- connector;
- interface_terminal;
- join_terminal.

4.2.727.1 connector

The connector specifies the role of the Printed_connector_template (see 4.2.726) for the Printed_connector_template_terminal_relationship.

4.2.727.2 interface_terminal

The interface_terminal specifies the role of the Printed_part_template_interface_terminal (see 4.2.732) for the Printed_connector_template_terminal_relationship.

4.2.727.3 join_terminal

The join_terminal specifies the role of the Printed_part_template_join_terminal (see 4.2.733) for the Printed_connector_template_terminal_relationship.

Constraints:

UR1: The combination of the join_terminal and the connector shall be unique within a population of Printed_connector_template_terminal_relationships.

UR2: The combination of the interface_terminal and the connector shall be unique within a population of Printed_connector_template_terminal_relationships.

WR1: The connector establishes the Part_template that has the contact with which the join_terminal is associated.

WR2: The connector establishes the Part_template that has the contact with which the interface_terminal is associated.

NOTE EXPRESS specification:

```

*)
ENTITY printed_connector_template_terminal_relationship
  SUBTYPE OF (managed_design_object);
    join_terminal      : printed_part_template_join_terminal;
    interface_terminal : printed_part_template_interface_terminal;
    connector          : printed_connector_template;
  UNIQUE
    UR1 : join_terminal, connector;
    UR2 : interface_terminal, connector;
END_ENTITY; -- printed_connector_template_terminal_relationship

--tt tc1 wd8 modified
( *

```

4.2.728 Printed_part_cross_section_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_cross_section_template is a type of Printed_part_template (see 4.2.730) that describes a cross sectional view of a Printed_part_template.

EXAMPLE 1 Figure 96 illustrates parameters, cross-sectional geometry, and materials needed to describe different transmission lines and related media.

The data associated with a Printed_part_cross_section_template are the following:

- maximum_width;
- minimum_width;
- transmission_line_model.

EXAMPLE 2 An example of Printed_part_cross_section_template is the two-dimensional shape based model in figure 96 for a regular transmission line structure. The cross sectional shape of the structure is invariant along the path of the center line of the structure. These structures include the definition of conductive Stratum_feature cross sections and the placement and location of dielectric Stratum in the Stratum_sequence. The width of each Stratum_feature is included.

4.2.728.1 maximum_width

The maximum_width specifies the maximum overall lateral Length_data_element (see 4.2.511) dimension through all the material of the Printed_part_cross_section_template.

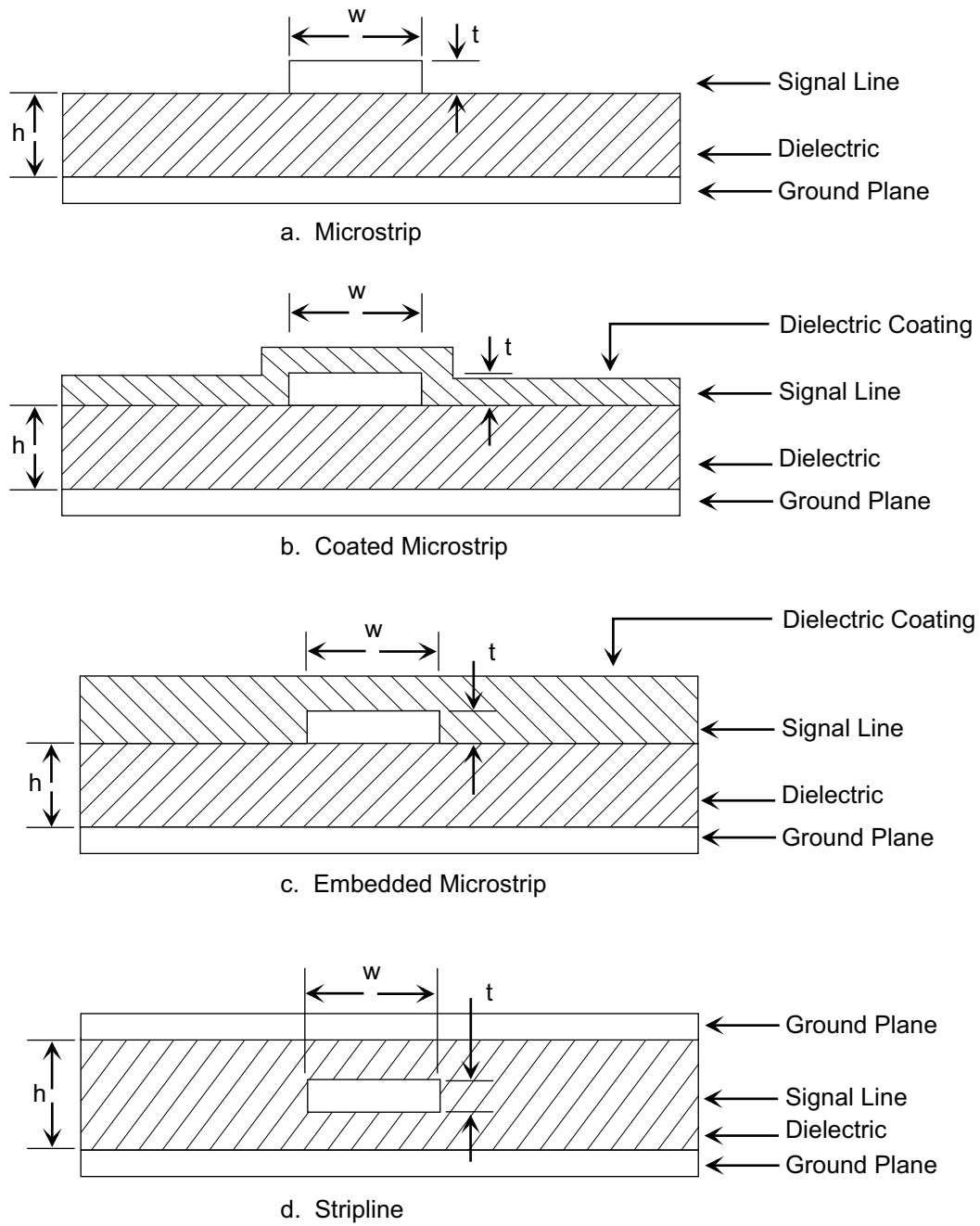


Figure 96 – Printed_part_cross_section_template

4.2.728.2 minimum_width

The minimum_width specifies the minimum overall lateral Length_data_element (see 4.2.511) dimension through all the material of the Printed_part_cross_section_template.

4.2.728.3 transmission_line_model

The transmission_line_model specifies the role of the Analytical_model (see 4.2.14) for the Printed_part_cross_section_template.

Constraints:

WR1: There shall be more than one terminal provided for a Printed_part_cross_section_template.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_cross_section_template
  SUBTYPE OF (part_template);
--tt tc1wd2 correction
  horizontal_material_link: OPTIONAL SET [1:?] OF
    printed_part_template_material_link;
  vertical_material_link : OPTIONAL SET [1:?] OF
    printed_part_template_material_link;
  transmission_line_model : analytical_model;
  maximum_transmission_line_characteristic : OPTIONAL SET [1:?] OF
    analytical_representation;
  minimum_transmission_line_characteristic : OPTIONAL SET [1:?] OF
    analytical_representation;
  maximum_width          : length_data_element;
  minimum_width          : length_data_element;
  start_vertex           : edge_segment_vertex;
  end_vertex             : edge_segment_vertex;
INVERSE
  using_terminal : SET [1:?] OF printed_part_cross_section_template_terminal FOR
    cross_section_definition;
WHERE
  WR1: SIZEOF( QUERY ( hml &#60;* horizontal_material_link |
    NOT (hml.precedent_material\shape_element.name in ['right']))) = 0;
  WR2: SIZEOF( QUERY ( hml &#60;* horizontal_material_link |
    NOT (hml.subsequent_material\shape_element.name in ['left']))) = 0;
  WR3: SIZEOF( QUERY ( hml &#60;* vertical_material_link |
    NOT (hml.precedent_material\shape_element.name in ['bottom']))) = 0;
  WR4: SIZEOF( QUERY ( hml &#60;* vertical_material_link |
    NOT (hml.precedent_material\shape_element.name in ['top']))) = 0;
  WR5: horizontal_material_link :&#60;&#62;: vertical_material_link;
  WR6: start_vertex :&#60;&#62;: end_vertex;
END_ENTITY; -- printed_part_cross_section_template

--tt tc1 wd8 modified
( *

```

4.2.729 Printed_part_cross_section_template_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_cross_section_template_terminal is a type of Printed_part_template_terminal (see 4.2.737) that is the terminal of the Printed_part_cross_section_template (see 4.2.728).

The data associated with a Printed_part_cross_section_template_terminal are the following:

— associated_definition.

4.2.729.1 associated_definition

The associated_definition specifies the role of the Printed_part_cross_section_template (see 4.2.728) for the Printed_part_cross_section_template_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_cross_section_template_terminal
  SUBTYPE OF (printed_part_template_terminal);
    cross_section_definition : printed_part_cross_section_template;
    material_to_left_of_terminal : OPTIONAL printed_part_template_material_link;
    material_to_right_of_terminal : OPTIONAL printed_part_template_material_link;
    material_to_top_of_terminal : OPTIONAL printed_part_template_material_link;
    material_to_bottom_of_terminal : OPTIONAL printed_part_template_material_link;
  UNIQUE
    UR1: cross_section_definition,
          material_to_left_of_terminal,
          material_to_right_of_terminal,
          material_to_top_of_terminal,
          material_to_bottom_of_terminal;
  WHERE
    WR1: (SELF\printed_part_template_terminal.connection_zone_category =
          curve_edge_segment) OR
          (SELF\printed_part_template_terminal.connection_zone_category =
          area_edge_segment);
    WR2: NOT(EXISTS (material_to_left_of_terminal) AND
              EXISTS (material_to_right_of_terminal)) OR
          (material_to_left_of_terminal :&#60;&#62;: material_to_right_of_terminal);
    WR3: NOT(EXISTS (material_to_left_of_terminal) AND
              EXISTS (material_to_top_of_terminal)) OR
          (material_to_left_of_terminal :&#60;&#62;: material_to_top_of_terminal);
    WR4: NOT(EXISTS (material_to_left_of_terminal) AND
              EXISTS (material_to_bottom_of_terminal)) OR
          (material_to_left_of_terminal :&#60;&#62;: material_to_bottom_of_terminal);
    WR5: NOT(EXISTS (material_to_right_of_terminal) AND
              EXISTS (material_to_top_of_terminal)) OR
          (material_to_right_of_terminal :&#60;&#62;: material_to_top_of_terminal);
    WR6: NOT(EXISTS (material_to_right_of_terminal) AND
              EXISTS (material_to_bottom_of_terminal)) OR
          (material_to_right_of_terminal :&#60;&#62;: material_to_bottom_of_terminal);
    WR7: NOT(EXISTS (material_to_top_of_terminal) AND
              EXISTS (material_to_bottom_of_terminal)) OR
          (material_to_top_of_terminal :&#60;&#62;: material_to_bottom_of_terminal);
  END_ENTITY; -- printed_part_cross_section_template_terminal

--tt tc1 wd9 modified
(

```

4.2.730 Printed_part_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template is a type of Part_template (see 4.2.620). Each Printed_part_template may be either one of or both of a Printed_connector_template (see 4.2.726) or a Printed_part_cross_section_template (see 4.2.728). A Printed_part_template is a Part_template that has a specified function. Material may be discontinuous in a Printed_part_template.

NOTE 1 An occurrence of this object is a Printed_component (see 4.2.721) that is realized during the manufacturing process for the printed circuit board.

NOTE 2 The functionality realized in an occurrence of a Printed part is determined by the shape of the occurrence and the material constituting the occurrence.

EXAMPLE 1 Figure 97 illustrates shape data associated with single layer Printed_part. This part of ISO 10303 supports complex, specific purpose, multi-layer Printed_part exchange using the design reuse concept with the Interconnect_module (see 4.2.440) being a design view that contains the full multi-layer definition of the part design. Individual elements of the part design on a layer basis would be related to individual elements in the design being created using the Component_make_from_relationship (see 4.2.126).

The data associated with a Printed_part_template are the following:

- access_mechanisms;
- characterized_material;
- cross_section;
- implemented_function.

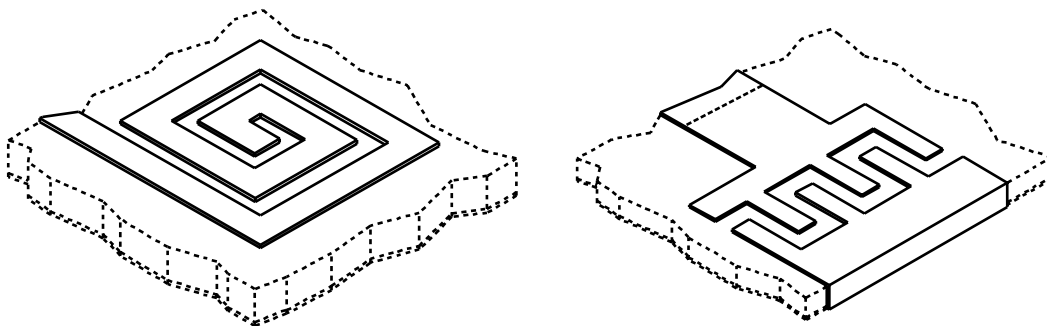


Figure 97 – Printed_part_template

4.2.730.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Printed_part_template is dependent on the existence of the Printed_part_template_terminal (see 4.2.737) that specifies the Printed_part_template as its associated_definition. There shall be two or more Printed_part_template_terminal for a Printed_part_template.

4.2.730.2 characterized_material

The characterized_material specifies the sequence of Printed_part_template_material (see 4.2.735) required to realize an implementation of a Printed_part_template, including conductive and dielectric materials. There shall be one or more Printed_part_template_material for a Printed_part_template, and the same characterized_material may occur more than once in the sequence.

NOTE The members of the sequence characterized_material need not be unique because the Printed_part_template is an incomplete set of information that is used as a reference in the construction of the complete product data for a design. The complete material shapes are not defined.

4.2.730.3 cross_section

The cross_section specifies the role of the Printed_part_cross_section_template (see 4.2.728) for the Printed_part_template. The cross_section need not be specified for a particular Printed_part_template.

4.2.730.4 implemented_function

The implemented_function specifies the role of the Functional_unit_usage_view (see 4.2.410) for the Printed_part_template.

Constraints:

WR1: The function this member of Printed_part_template implements shall be the associated function for the functional terminals mapped to the terminals of this member of Printed_part_template.

NOTE The Printed_part_template will have two product categories populated: one for the Part_template and one for the Printed_part_template. There is no conflict between these two categories.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template
  SUPERTYPE OF (ONEOF(layout_macro_floor_plan_template,printed_connector_template))
  SUBTYPE OF (part_template);
--tt tc1wd2 correct list error, make vertical link set
--tt tc1wd2 change attribute name for clarity.
  of_stratum_technology : OPTIONAL stratum_technology;
  material_stack : OPTIONAL SET [1:?] OF stratum_technology_link;
  printed_part_template_stack : OPTIONAL SET [1:?] OF printed_part_template_link;
  implemented_function : functional_unit_usage_view;
DERIVE
  access_mechanisms : SET [1:?] OF printed_part_template_terminal :=
    raw_access_mechanisms +
    printed_part_template_access_mechanisms(printed_part_template_stack);
INVERSE
--tt tc1 wd9 modify for stacked printed_part_template
  raw_access_mechanisms : SET [0:?] OF printed_part_template_terminal FOR
    associated_definition;
WHERE
  WR1: ((EXISTS (of_stratum_technology) XOR
    EXISTS (material_stack)) AND
    (EXISTS (of_stratum_technology) XOR
    EXISTS (printed_part_template_stack)) AND
    (EXISTS (material_stack) XOR
    EXISTS (printed_part_template_stack))) OR
    ('AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN TYPEOF(SELF));
  WR2: (NOT (EXISTS(material_stack))
    OR (SIZEOF(QUERY ( temp &#60;* material_stack |
    EXISTS (temp.precedent_stratum_technology_link))) =
    SIZEOF(material_stack) -1)) OR
    ('AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN TYPEOF(SELF));
--tt tc1 wd9 added following
  WR3: (SIZEOF (QUERY( prpc &#60;*
    USEDIN(associated_definition\ee_product_definition.version.versioned_item,
    'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') |
    (prpc.name = 'printed part')
    )) &#62;= 0) OR
    ('AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN TYPEOF(SELF));
  WR4: (EXISTS(printed_part_template_stack) XOR EXISTS(raw_access_mechanisms))
    OR
    ('AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN TYPEOF(SELF));
END_ENTITY; -- printed_part_template

( *
```

4.2.731 Printed_part_template_connected_terminals_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_connected_terminals_definition is a type of Managed_design_object (see 4.2.524).
A Printed_part_template_connected_terminals_definition is the association between two or more Printed_part_template_terminal (see 4.2.737) that are connected.

The data associated with a Printed_part_template_connected_terminals_definition are the following:

- connected_terminals;
- name.

4.2.731.1 connected_terminals

The connected_terminals specify the Printed_part_template_terminal (see 4.2.737) for the Printed_part_template_connected_terminals_definition. There shall be two or more Printed_part_template_terminal for the Printed_part_template_connected_terminals_definition.

4.2.731.2 name

The name identifies the Printed_part_template_connected_terminals_definition.

Constraints:

UR1: The name shall be unique within a population of Printed_part_template_connected_terminals_definitions.

NOTE EXPRESS specification:

```
*)
ENTITY printed_part_template_connected_terminals_definition
  SUBTYPE OF (managed_design_object);
    connected_terminals : SET [2:?] OF printed_part_template_terminal;
    name                  : ee_name;
  UNIQUE
    UR1 : name;
END_ENTITY; -- printed_part_template_connected_terminals_definition

(*
```

4.2.732 Printed_part_template_interface_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_interface_terminal is a type of Printed_part_template_terminal (see 4.2.737), an occurrence of which is an external access to the printed circuit board of which it is a component.

The data associated with a Printed_part_template_interface_terminal are the following:

— external_connector_function.

4.2.732.1 external_connector_function

The external_connector_function is an inverse relationship that specifies the Printed_connector_template_terminal_relationship (see 4.2.727) that specifies the Printed_part_template_interface_terminal as its interface_terminal. There shall be one or more Printed_connector_template_terminal_relationship for a Printed_part_template_interface_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_interface_terminal
  SUBTYPE OF (printed_part_template_terminal);
  SELF\printed_part_template_terminal.associated_definition :
                                     printed_connector_template;
  INVERSE
    external_connector_function : SET [1:?] OF
                                     printed_connector_template_terminal_relationship FOR
                                     interface_terminal;
END_ENTITY; -- printed_part_template_interface_terminal

( *
```

4.2.733 Printed_part_template_join_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_join_terminal is a type of Printed_part_template_terminal (see 4.2.737) an occurrence of which provides access to the functionality of the Printed_part_template (see 4.2.730) from within a printed circuit board.

NOTE Join is a concept that explicitly states the purpose is for completing the network connections internal to a physical design.

The data associated with a Printed_part_template_join_terminal are the following:

— internal_connector_function.

4.2.733.1 internal_connector_function

The internal_connector_function is an inverse relationship that specifies the Printed_connector_template_terminal_relationship (see 4.2.727) that specifies the Printed_part_template_join_terminal as its join_terminal. There shall be no more than one Printed_connector_template_terminal_relationship for a Printed_part_template_join_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_join_terminal
  SUBTYPE OF (printed_part_template_terminal);
  INVERSE
    internal_connector_function : SET [0:1] OF
      printed_connector_template_terminal_relationship FOR
        join_terminal;
END_ENTITY; -- printed_part_template_join_terminal

--tt tc1 wd9 added
( *
```

4.2.734 Printed_part_template_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_link is a type of Stratum_technology_link (see 4.2.872).

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_link
  SUBTYPE OF (stratum_technology_link);
  precedent_printed_part_template : printed_part_template;
  subsequent_printed_part_template : printed_part_template;
  INVERSE
    associated_printed_part_template : printed_part_template FOR
      printed_part_template_stack;
  WHERE
    WR1: EXISTS(precedent_printed_part_template.
      of_stratum_technology);
    WR2: EXISTS(subsequent_printed_part_template.
      of_stratum_technology);
  --tt tc1 wd9 modified
  WR3: precedent_printed_part_template.of_stratum_technology =
    SELF\stratum_technology_link.precedent_stratum_technology;
  --tt tc1 wd9 modified
  WR4: subsequent_printed_part_template.of_stratum_technology =
    SELF\stratum_technology_link.subsequent_stratum_technology;
END_ENTITY; -- printed_part_template_link

( *
```

4.2.735 Printed_part_template_material

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_material is a type of Managed_design_object (see 4.2.524). A Printed_part_template_material is the Ee_material (see 4.2.318) characterized in Printed_part_cross_section_template (see 4.2.728), for the realization of the Stratum (see 4.2.845) containing part of the Printed_component (see 4.2.721) during fabrication of the Interconnect_module (see 4.2.440).

NOTE A Printed_component may span more than one Stratum, as defined by the Printed_part_template (see 4.2.730).

The data associated with a Printed_part_template_material are the following:

- maximum_thickness;
- minimum_thickness;
- required_material.

4.2.735.1 maximum_thickness

The maximum_thickness specifies the maximum Length_data_element (see 4.2.511) for the thickness of the Printed_part_template_material.

4.2.735.2 minimum_thickness

The minimum_thickness specifies the minimum Length_data_element (see 4.2.511) for the thickness of the Printed_part_template_material.

4.2.735.3 required_material

The required_material specifies the role of the Ee_material (see 4.2.318) for the Printed_part_template_material.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_material
  SUBTYPE OF (managed_design_object);
--tt tc1 correction
--  maximum_thickness : length_data_element;
--  required_material : ee_material;
--  minimum_thickness : length_data_element;
  maximum_width      : OPTIONAL length_data_element;
  minimum_width      : OPTIONAL length_data_element;
  required_material : stratum_technology;
  INVERSE
    boundary :
SET [0:4] OF template_material_cross_section_boundary FOR
associated_definition;
--tt tc1wd8 correction added
WHERE
  WR1: SIZEOF(QUERY( temp &#60;* (USEDIN(SELF,
    'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' +
    'ASSOCIATED_DEFINITION'))| temp\shape_element.name = 'top')) &#60;= 1;
  WR2: SIZEOF(QUERY( temp &#60;* (USEDIN(SELF,
    'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' +
    'ASSOCIATED_DEFINITION'))| temp\shape_element.name = 'bottom')) &#60;= 1;
  WR3: SIZEOF(QUERY( temp &#60;* (USEDIN(SELF,
    'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' +
    'ASSOCIATED_DEFINITION'))| temp\shape_element.name = 'left')) &#60;= 1;
  WR4: SIZEOF(QUERY( temp &#60;* (USEDIN(SELF,
    'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' +
    'ASSOCIATED_DEFINITION'))| temp\shape_element.name = 'right')) &#60;= 1;
END_ENTITY; -- printed_part_template_material

--tt tc1 correction add
(

```

4.2.736 Printed_part_template_material_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_material_link is a type of Managed_design_object (see 4.2.524).

The data associated with a Printed_part_template_material_link are the following:

- precedent_material;
- subsequent_material.

4.2.736.1 precedent_material

The precedent_material specifies a Template_material_cross_section_boundary (see 4.2.888) for the Printed_part_template_material_link.

4.2.736.2 subsequent_material

The subsequent_material specifies a Template_material_cross_section_boundary (see 4.2.888) for the Printed_part_template_material_link.

Constraints:

UR1: The combination of precedent_material and subsequent_material shall be unique within a population of Printed_part_template_material_link.

WR1: The precedent_material shall not be the subsequent_material within a population of Printed_part_template_material_link.

WR2: The name attribute inherited from Shape_element (see 4.2.810) for the precedent_material shall be one of "bottom" or "right" within a population of Template_material_cross_section_boundary.

WR3: The name attribute inherited from Shape_element (see 4.2.810) for the subsequent_material shall be one "top" or "left" within a population of Template_material_cross_section_boundary.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_material_link
  SUBTYPE OF (managed_design_object);
    precedent_material : template_material_cross_section_boundary;
    subsequent_material : template_material_cross_section_boundary;
  UNIQUE
    UR1: precedent_material, subsequent_material;
  WHERE
    WR1: precedent_material :&#60;&#62;: subsequent_material;
    --tt tc1 wd8 added
    WR2: precedent_material\shape_element.name in ['bottom', 'right'];
    WR3: subsequent_material\shape_element.name in ['top', 'left'];
  END_ENTITY; -- printed_part_template_material_link

( *
```

4.2.737 Printed_part_template_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Printed_part_template_terminal` is a type of `Managed_design_object` (see 4.2.524). A `Printed_part_template_terminal` is a `Printed_part_cross_section_template_terminal` (see 4.2.729) or additionally one of a `Printed_part_template_interface_terminal` (see 4.2.732) or a `Printed_part_template_join_terminal` (see 4.2.733). A `Printed_part_template_terminal` is a terminal of a `Printed_part_template` (see 4.2.730).

The data associated with a `Printed_part_template_terminal` are the following:

- `associated_definition`;
- `connection_requirement`;
- `connection_zone_category`;
- `feature_shape`;
- `functional_usage_view_terminal`;
- `name`;
- `terminal_connection_zone`.

4.2.737.1 associated_definition

The `associated_definition` specifies the role of the `Printed_part_template` (see 4.2.730) for the `Printed_part_template_terminal`.

4.2.737.2 connection_requirement

The `connection_requirement` is an inverse relationship that specifies that the existence of the `Printed_part_template_terminal` is dependent on the existence of the `Printed_part_template_connected_terminals_definition` (see 4.2.731) that specifies the `Printed_part_template_terminal` as its `connected_terminals`. There shall be no more than one `Printed_part_template_connected_terminals_definition` for a `Printed_part_template_terminal`.

4.2.737.3 connection_zone_category

The `connection_zone_category` specifies either an `area_edge_segment`, a `curve_edge_segment`, a `surface_area`, or a `surface_point` for the `Printed_part_template_terminal`.

4.2.737.4 feature_shape

The `feature_shape` specifies the role of the `Feature_shape_occurrence` (see ??) for the `Printed_part_template_terminal`. There shall be one or more `Feature_shape_occurrence` for the `Printed_part_template_terminal`.

4.2.737.5 functional_usage_view_terminal

The functional_usage_view_terminal is an inverse relationship that specifies the Functional_usage_view_to_printed_part_template_terminal_assignment (see 4.2.414) that specifies the Printed_part_template_terminal as its physical_usage_view_terminal. There shall be no more than one Functional_usage_view_to_printed_part_template_terminal_assignment for a Printed_part_template_terminal.

4.2.737.6 name

The name identifies the descriptive label for the Printed_part_template_terminal.

4.2.737.7 terminal_connection_zone

The terminal_connection_zone specifies the role of the Connection_zone (see 4.2.185) for the Printed_part_template_terminal. There may be one or more Connection_zone for a Printed_part_template_terminal. The terminal_connection_zone is optional and need not be supplied if the source system does not contain the data. If the source system contains the data, then the terminal_connection_zone shall be provided. The Feature_shape_occurrence (see ??) specified by the terminal or mounting feature may be the same Feature_shape_occurrence that is specified by the associated Connection_zone (see 4.2.185). In this case the whole terminal is specified as the Connection_zone.

Constraints:

UR1: The combination of the name and the associated_definition shall be unique within a population of Printed_part_template_terminals.

WR1: The points comprising the attachment surface must reside in the shape of the electrical device being accessed.

WR2: The points comprising the terminal_connection_zone shall be defined in the coordinate system of the device being accessed via the terminal_connection_zone.

NOTE EXPRESS specification:

```

*)
ENTITY printed_part_template_terminal
  ABSTRACT SUPERTYPE OF (
    printed_part_cross_section_template_terminal
    ANDOR (ONEOF (printed_part_template_join_terminal,
      printed_part_template_interface_terminal)))
  SUBTYPE OF (managed_design_object);
  associated_definition      : printed_part_template;
  connection_zone_category : printed_part_template_terminal_class;
  terminal_connection_zone : OPTIONAL SET [1:?] OF connection_zone;
  name                      : ee_name;
  INVERSE
    connection_requirement      :
  SET [0:1] OF printed_part_template_connected_terminals_definition FOR
  connected_terminals;
  functional_usage_view_terminal :
  SET [0:1] OF functional_usage_view_to_printed_part_template_terminal_assignment FOR
  physical_usage_view_terminal;
  feature_shape                : SET [1:?] OF usage_concept_usage_relationship FOR
  associated_usage;
  UNIQUE
    UR1 : name, associated_definition;
END_ENTITY; -- printed_part_template_terminal

( *

```

4.2.738 Printed_part_template_terminal_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Printed_part_template_terminal_relationship is a type of Managed_design_object (see 4.2.524). A Printed_part_template_terminal_relationship is an association between two Printed_part_template_terminal (see 4.2.737) in a Printed_part_template (see 4.2.730).

The data associated with a Printed_part_template_terminal_relationship are the following:

- related_printed_part_terminal;
- relating_printed_part_terminal.

4.2.738.1 related_printed_part_terminal

The related_printed_part_terminal specifies the dependent Printed_part_template_terminal (see 4.2.737) in the Printed_part_template_terminal_relationship.

4.2.738.2 relating_printed_part_terminal

The relating_printed_part_terminal specifies the independent Printed_part_template_terminal (see 4.2.737) in the Printed_part_template_terminal_relationship.

NOTE EXPRESS specification:

```
*)
ENTITY printed_part_template_terminal_relationship
  SUBTYPE OF (managed_design_object);
    related_printed_part_terminal : printed_part_template_terminal;
    relating_printed_part_terminal : printed_part_template_terminal;
END_ENTITY; -- printed_part_template_terminal_relationship

( *
```

4.2.739 Probe_access_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Probe_access_area is a type of Laminate_component (see 4.2.483). A Probe_access_area may be an Internal_probe_access_area (see 4.2.475). A Probe_access_area is a feature incorporated into the design of an Interconnect_module (see 4.2.440) for the purpose of measuring electrical characteristics of the design. It is realized by an exposed surface of conducting material that is in physical contact with the Conductor (see 4.2.176) or Conductive_filled_area (see 4.2.170) upon which the measurement is to be made.

EXAMPLE 1 Figure 98 illustrates a detail of a layout drawing that shows a probe area in the layout. Sufficient detail may be included in the design to evaluate accurately probe placement and volume restrictions.

The data associated with a Probe_access_area are the following:

- probe_contact_zone;
- probed_layout_item.

4.2.739.1 probe_contact_zone

The probe_contact_zone specifies the role of the Connection_zone (see 4.2.185) for the Probe_access_area.

4.2.739.2 probed_layout_item

The probed_layout_item specifies the role of the Stratum_feature (see 4.2.851) for the Probe_access_area.

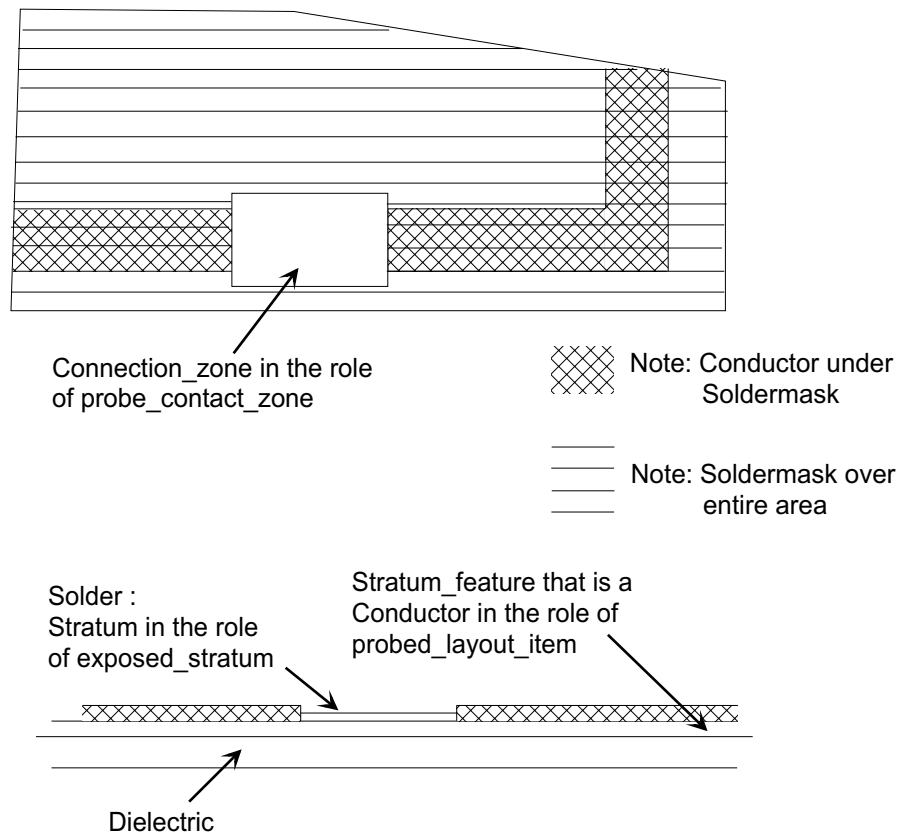


Figure 98 – Probe_access_area

NOTE EXPRESS specification:

```

*)
ENTITY probe_access_area
  SUPERTYPE OF (internal_probe_access_area)
  SUBTYPE OF (laminate_component);
    probed_layout_item : stratum_feature;
    probe_contact_zone : connection_zone;
END_ENTITY; -- probe_access_area

( *
```

4.2.740 Process_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Process_specification is a type of Ee_specification (see 4.2.338) that identifies and describes an activity to be performed on an Ee_product (see 4.2.319) or Ee_material (see 4.2.318). A Process_specification may be an Assembly_technology_specification (see 4.2.55), a Fabrication_technology_specification (see 4.2.369) or a Lead_form_specification (see 4.2.510).

EXAMPLE 1 Figure 99 illustrates a detail from a Process_specification for an etching process.

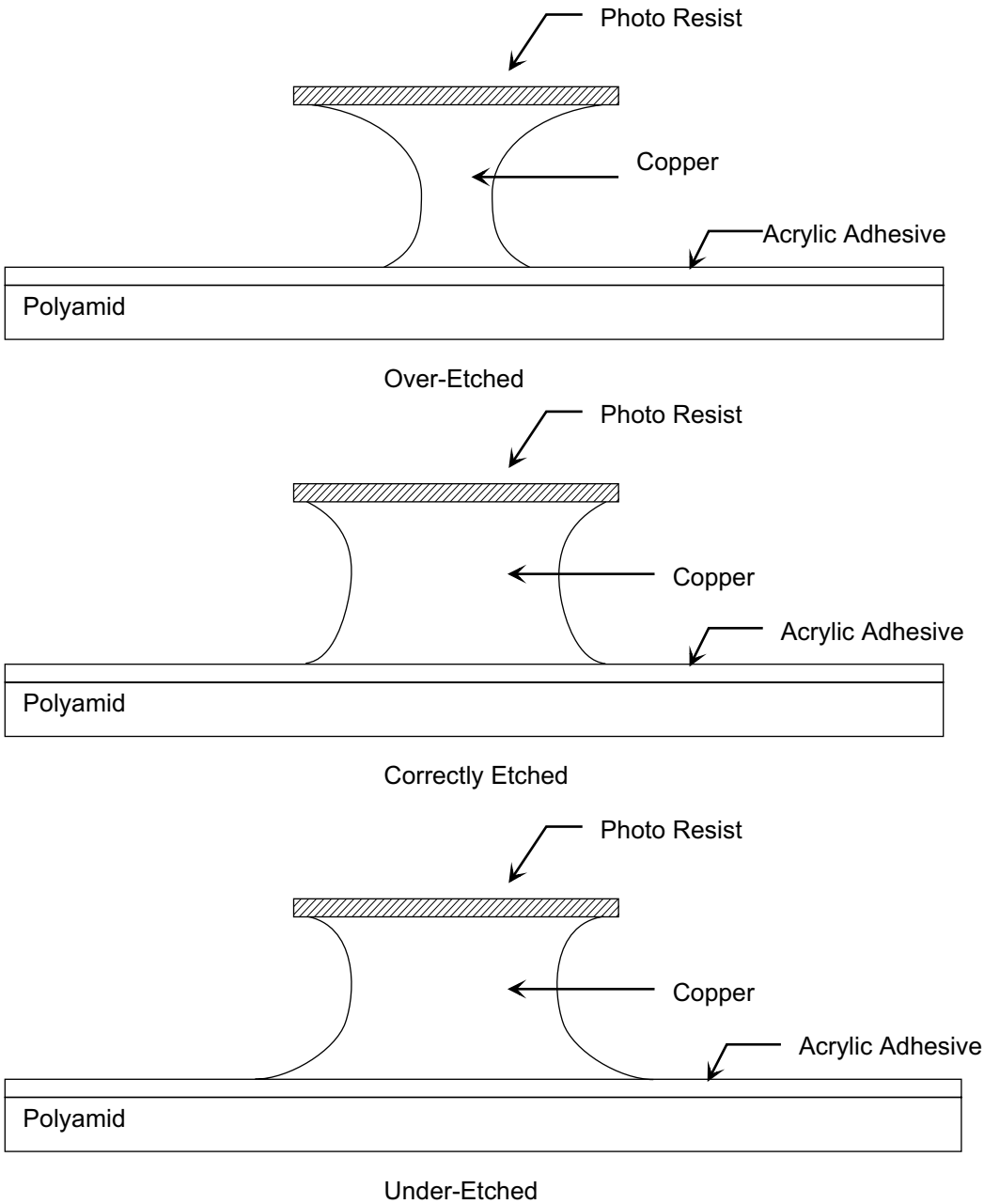


Figure 99 – Process_specification

NOTE EXPRESS specification:

```

*)
ENTITY process_specification
  SUPERTYPE OF (ONEOF (assembly_technology_specification,
    fabrication_technology_specification, lead_form_specification))
  SUBTYPE OF (ee_specification);
END_ENTITY; -- process_specification

( *

```

4.2.741 Product_association

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Product_association is a type of Managed_design_object (see 4.2.524). A Product_association is a usage constraint relationship among some parts that may be used in a design. If the relating part is used in a design then the related parts must also be incorporated in the design and in a prescribed fashion.

The data associated with a Product_association are the following:

- constraint_source;
- related_product;
- relating_product.

4.2.741.1 constraint_source

The constraint_source specifies the Ee_specification (see 4.2.338) that describes the manner in which Ee_product_version (see 4.2.328) that participate in the Product_association are to be used in the design.

4.2.741.2 related_product

The related_product specifies the Ee_product_version (see 4.2.328) that must be incorporated in the design when the relating_product is used in the design for the Product_association. There shall be one or more Ee_product_version for the Product_association.

4.2.741.3 relating_product

The relating_product specifies the Ee_product_version (see 4.2.328) that, when incorporated in the design, requires the incorporation of one or more related_product to be incorporated in the design to realize the desired functionality of the relating_product for the Product_association.

Constraints:

UR1: The constraint_source shall be unique within a population of Product_associations.

WR1: An Ee_product_version shall not be an ancestor or descendent of itself in a population of this object.

NOTE EXPRESS specification:

```

*)
ENTITY product_association
  SUBTYPE OF (managed_design_object);
    relating_product : ee_product_version;
    related_product  : SET [1:?] OF ee_product_version;
    constraint_source : ee_specification;
  UNIQUE
    UR1 : constraint_source, relating_product;
END_ENTITY; -- product_association

( *
```

4.2.742 Profile_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Profile_boundary_definition is a type of Boundary_zone_definition_with_specified_size (see 4.2.79) that characterizes a Tolerance_zone_boundary (see 4.2.915) of a Tolerance_zone (see 4.2.913) that is defined by either a Profile_of_any_line_tolerance (see 4.2.744) or a Profile_of_any_surface_tolerance (see 4.2.747). A Profile_boundary_definition may be a Profile_boundary_definition_with_offsets (see 4.2.743).

The data associated with a Profile_boundary_definition are the following:

- basic_profile;
- defined_zone_boundary.

4.2.742.1 basic_profile

The basic_profile specifies the True_profile (see 4.2.934) that characterizes the theoretically exact profile of the toleranced Physical_feature_or_part_template (see 4.2.658) along which the Tolerance_zone_opposing_boundary_set (see 4.2.926) is disposed.

4.2.742.2 defined_zone_boundary

The `defined_zone_boundary` specifies the `Tolerance_zone_opposing_boundary_set` (see 4.2.926) that is characterized by the `Profile_boundary_definition`. If a `Tolerance_zone_implicit_boundary_set` is specified as the `defined_zone_boundary`, the `Tolerance_zone_boundary` (see 4.2.915) is equally disposed about the `True_profile` (see 4.2.934).

Constraints:

WR1: The `Tolerance_zone_definition` specified as the `zone_definition` shall be specified as the `tolerance_zone_or_boundary_definition` by either a `Profile_of_any_line_tolerance` or a `Profile_of_any_surface_tolerance`.

NOTE EXPRESS specification:

```

*)
ENTITY profile_boundary_definition
  SUPERTYPE OF (profile_boundary_definition_with_offsets)
  SUBTYPE OF (boundary_zone_definition_with_specified_size);
    basic_profile : true_profile;
    SELF\tolerance_zone_boundary_definition.defined_zone_boundary :
tolerance_zone_opposing_boundary_set;
END_ENTITY; -- profile_boundary_definition

( *
```

4.2.743 Profile_boundary_definition_with_offsets

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Profile_boundary_definition_with_offsets` is a type of `Profile_boundary_definition` (see 4.2.742) in which the offset of each `Tolerance_zone_boundary_member` (see 4.2.917) relative to the `True_profile` (see 4.2.934) is explicitly defined.

The data associated with a `Profile_boundary_definition_with_offsets` are the following:

- `boundary_offsets`;
- `defined_zone_boundary`.

4.2.743.1 boundary_offsets

The `boundary_offsets` specifies the `Tolerance_zone_boundary_member_offset` (see 4.2.918) for each `Tolerance_zone_boundary_member` (see 4.2.917) of the `Tolerance_zone_explicit_opposing_boundary_set` (see 4.2.923) specified as the `defined_zone_boundary`. Each `Tolerance_zone_boundary_member_offset` characterizes the offset of a `Tolerance_zone_boundary_member` from the `True_profile` (see 4.2.934). There shall be exactly two `Tolerance_zone_boundary_member_offset` specified as the `boundary_offsets` for the `Profile_boundary_definition_with_offsets`.

4.2.743.2 defined_zone_boundary

The `defined_zone_boundary` specifies the `Tolerance_zone_explicit_opposing_boundary_set` (see 4.2.923) that is characterized by the `Profile_boundary_definition_with_offsets`.

Constraints:

WR1: The sum of each `Length_data_element` specified as the `offset_value` by the two `Tolerance_zone_boundary_member` that are specified as the `boundary_offsets` shall equal the `Length_data_element` specified by the `tolerance_value` of the `Boundary_size_characteristic` specified as the `specified_boundary_size`.

WR2: The value of the label specified by the `type_of_offset` of one of the `Tolerance_zone_boundary_member_offsets` specified by the `boundary_offsets` shall be "away_from_material".

WR3: The value of the label specified by the `type_of_offset` of one of the `Tolerance_zone_boundary_member_offsets` specified by the `boundary_offsets` shall be "towards offset".

NOTE EXPRESS specification:

```

*)
ENTITY profile_boundary_definition_with_offsets
  SUBTYPE OF (profile_boundary_definition);
    SELF\tolerance_zone_boundary_definition.defined_zone_boundary :
tolerance_zone_explicit_opposing_boundary_set;
    boundary_offsets :
SET [2:2] OF tolerance_zone_boundary_member_offset;
END_ENTITY; -- profile_boundary_definition_with_offsets

( *
```

4.2.744 Profile_of_any_line_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Profile_of_any_line_tolerance` is a type of `Geometric_tolerance` (see 4.2.419). A `Profile_of_any_line_tolerance` is either a `Profile_of_any_line_tolerance_without_referenced_datum_system` (see 4.2.746) or a `Profile_of_any_line_tolerance_with_referenced_datum_system` (see 4.2.745). A `Profile_of_any_line_tolerance` may also be a `Composite_linear_profile_refinement_tolerance` (see 4.2.161). A `Profile_of_any_line_tolerance` participates in the specification of a design requirement on the form, and possibly on the orientation, size, and location of line elements of a surface. The `Profile_of_any_line_tolerance` is similar to the `Profile_of_any_surface_tolerance` (see 4.2.747) in that the `Tolerance_zone` (see 4.2.913) is based on the `True_profile` (see 4.2.934). However, with a `Profile_of_any_line_tolerance`, each line element of the surface must lie within its associated `Tolerance_zone`, where the line elements are the cross-sections of the tolerated surface that are parallel to the specified `Viewing_plane` (see 4.2.956). If the design requirements are such that the `Tolerance_zone_boundary` (see 4.2.915) is unequally disposed about the `True_profile`, either unilaterally or bilateral, a `Profile_boundary_definition_with_offsets` (see 4.2.743) shall be used to specify this requirement.

NOTE Profile of any line tolerance is defined in terms of tolerance zone definitions in 14.5 of ISO 1101.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` shall be a `Physical_feature_or_part_template`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

WR2: The `Profile_of_any_line_tolerance` shall not be a `Geometric_tolerance_with_applied_material_condition_property`.

WR3: The `Profile_of_any_line_tolerance` shall specify as the `tolerance_zone_or_boundary_definition` a `Tolerance_zone_definition` that specifies a `Profile_boundary_definition` as its `boundary_definition`.

IP1: The form of the tolerated `Shape_element` shall be consistent with the form of a shape element for which the application of a profile of any line tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY profile_of_any_line_tolerance
  ABSTRACT SUPERTYPE OF (ONEOF (
    profile_of_any_line_tolerance_with_referenced_datum_system,
    profile_of_any_line_tolerance_without_referenced_datum_system)
  ANDOR composite_linear_profile_refinement_tolerance)
  SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- profile_of_any_line_tolerance

( *
```

4.2.745 Profile_of_any_line_tolerance_with_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Profile_of_any_line_tolerance_with_referenced_datum_system` is a type of `Geometric_tolerance_-with_referenced_datum_system` (see 4.2.422) and a type of `Profile_of_any_line_tolerance` (see 4.2.744) in which the line elements of the tolerated `Physical_feature_or_part_template` (see 4.2.658) are oriented and/or located in relation to a `Datum_system` (see 4.2.236). A `Profile_of_any_line_tolerance_with_referenced_datum_system` may be a `Composite_linear_profile_locating_tolerance` (see 4.2.160).

NOTE EXPRESS specification:

```

*)
ENTITY profile_of_any_line_tolerance_with_referenced_datum_system
  SUPERTYPE OF (composite_linear_profile_locating_tolerance)
  SUBTYPE OF (profile_of_any_line_tolerance,
    geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- profile_of_any_line_tolerance_with_referenced_datum_system

( *
```

4.2.746 **Profile_of_any_line_tolerance_without_referenced_datum_system**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Profile_of_any_line_tolerance_without_referenced_datum_system` is a type of `Geometric_tolerance_-without_referenced_datum_system` (see 4.2.423) and a type of `Profile_of_any_line_tolerance` (see 4.2.744) in which the line elements of the tolerated `Physical_feature_or_part_template` (see 4.2.658) are not oriented and/or located in relation to a `Datum_system` (see 4.2.236). That is, it is a `Profile_of_any_line_-tolerance` that specifies design requirements on the profile of line elements of the `Shape_element` (see 4.2.810) to which the `Profile_of_any_line_tolerance_without_referenced_datum_system` is applied, and in which the line elements are oriented parallel to a specified `Viewing_plane` (see 4.2.956) as opposed to oriented relative to a referenced `Datum_system`.

Constraints:

WR1: The `Tolerance_zone_definition` specified as the `tolerance_zone_or_boundary_definition` shall be a `Tolerance_zone_definition_with_specified_orientation` that specifies a `Viewing_plane_based_orientation` as its `zone_orientation`.

NOTE EXPRESS specification:

```

*)
ENTITY profile_of_any_line_tolerance_without_referenced_datum_system
  SUBTYPE OF (profile_of_any_line_tolerance,
    geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- profile_of_any_line_tolerance_without_referenced_datum_system

( *
```

4.2.747 Profile_of_any_surface_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Profile_of_any_surface_tolerance is a type of Geometric_tolerance (see 4.2.419) that specifies or partially specifies the design requirement on the form and possibly on the orientation, size, and location of a surface. A Profile_of_any_surface_tolerance is either a Profile_of_any_surface_tolerance_with_referenced_datum_system (see 4.2.748) or a Profile_of_any_surface_tolerance_without_referenced_datum_system (see 4.2.749). A Profile_of_any_surface_tolerance may also be a Composite_surface_profile_refinement_tolerance (see 4.2.166). The Profile_of_any_surface_tolerance defines a uniform Tolerance_zone_boundary (see 4.2.915) along the true profile of the surface within which the actual surface must lie. If the design requirements are such that the Tolerance_zone_boundary is unequally disposed about the True_profile (see 4.2.934), either unilaterally or bilateral, a Profile_boundary_definition_with_offsets (see 4.2.743) shall be used to specify this requirement.

NOTE Profile of any surface tolerance is defined in terms of tolerance zone definitions in 14.6 of ISO 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Physical_feature_or_part_template.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: The Profile_of_any_surface_tolerance shall not be a Geometric_tolerance_with_applied_material_condition_property.

WR3: The Profile_of_any_surface_tolerance shall specify as the tolerance_zone_or_boundary_definition a Tolerance_zone_definition that specifies a Profile_boundary_definition as its boundary_definition.

NOTE EXPRESS specification:

```

*)
ENTITY profile_of_any_surface_tolerance
  ABSTRACT SUPERTYPE OF (ONEOF (
    profile_of_any_surface_tolerance_with_referenced_datum_system,
    profile_of_any_surface_tolerance_without_referenced_datum_system)
  ANDOR composite_surface_profile_refinement_tolerance)
  SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- profile_of_any_surface_tolerance

( *

```

4.2.748 Profile_of_any_surface_tolerance_with_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Profile_of_any_surface_tolerance_with_referenced_datum_system is a type of Geometric_tolerance_-with_referenced_datum_system (see 4.2.422) and a type of Profile_of_any_surface_tolerance (see 4.2.747). That is, it is a type of Profile_of_any_surface_tolerance in which the Physical_feature_or_part_template (see 4.2.658) to which it is applied is oriented and/or located in relation to a Datum_system (see 4.2.236). A Profile_of_any_surface_tolerance_with_referenced_datum_system may be a Composite_surface_profile_locating_tolerance (see 4.2.165).

NOTE EXPRESS specification:

```

*)
ENTITY profile_of_any_surface_tolerance_with_referenced_datum_system
  SUPERTYPE OF (composite_surface_profile_locating_tolerance)
  SUBTYPE OF (profile_of_any_surface_tolerance,
    geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- profile_of_any_surface_tolerance_with_referenced_datum_system

( *

```

4.2.749 Profile_of_any_surface_tolerance_without_referenced_datum_system

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Profile_of_any_surface_tolerance_without_referenced_datum_system is a type of Geometric_tolerance_without_referenced_datum_system (see 4.2.423) and a type of Profile_of_any_surface_tolerance (see 4.2.747). That is, it is a Profile_of_any_surface_tolerance in which the Physical_feature_or_part_template (see 4.2.658) to which the Profile_of_any_surface_tolerance_without_referenced_datum is applied is not oriented or located in relation to a Datum_system (see 4.2.236).

NOTE EXPRESS specification:

```

*)
ENTITY profile_of_any_surface_tolerance_without_referenced_datum_system
  SUBTYPE OF (profile_of_any_surface_tolerance,
    geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- profile_of_any_surface_tolerance_without_referenced_datum_system

( *
```

4.2.750 Profile_related_positional_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Profile_related_positional_boundary is a type of Positional_boundary (see 4.2.698) that is a theoretical boundary that is defined by a Boundary_based_position_tolerance (see 4.2.77) in conjunction with a Profile_of_any_surface_tolerance (see 4.2.747). The Profile_of_any_surface_tolerance defines a Tolerance_zone (see 4.2.913) that is bounded by two Tolerance_zone_boundary_member (see 4.2.917). The Profile_related_positional_boundary is offset from one of these Tolerance_zone_boundary_member. The particular Tolerance_zone_boundary_member and the direction of offset are determined by the material conditions specified by the Boundary_based_position_tolerance. If the Boundary_based_position_tolerance is specified on a "maximum material principle" basis, then the Tolerance_zone_boundary_member from which the Profile_related_positional_boundary is offset is the one that is offset from the True_profile (see 4.2.934)"away from material". Similarly, if the Boundary_based_position_tolerance is specified on a "least material requirement" basis, then the Tolerance_zone_boundary_member from which the Profile_related_positional_boundary is offset is the one that is offset from the True_profile "towards material". The Profile_related_positional_boundary is offset from the Tolerance_zone_boundary_member in the same direction that the Tolerance_zone_boundary_member is offset from the True_profile.

The data associated with a Profile_related_positional_boundary are the following:

— associating_relationship.

4.2.750.1 associating_relationship

The associating_relationship is an inverse relationship that specifies that the existence of the Profile_related_positional_boundary is dependent on the existence of the Positional_boundary_and_profile_boundary_member_relationship (see 4.2.699) that specifies the Profile_related_positional_boundary as its offset_positional_boundary.

Constraints:

WR1: The Positional_boundary_definition specified as the boundary_definition shall be a Profile_related_positional_boundary_definition.

NOTE The attribute `boundary_definition` is declared in the `Positional_boundary` entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY profile_related_positional_boundary
  SUBTYPE OF (positional_boundary);
  INVERSE
    associating_relationship :
    positional_boundary_and_profile_boundary_member_relationship FOR
    offset_positional_boundary;
END_ENTITY; -- profile_related_positional_boundary

( *
```

4.2.751 Profile_related_positional_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Profile_related_positional_boundary_definition` is a type of `Positional_boundary_definition` (see 4.2.700) that characterizes a `Profile_related_positional_boundary` (see 4.2.750). A `Profile_related_positional_boundary_definition` is defined by a `Boundary_based_position_tolerance` (see 4.2.77) in conjunction with a `Profile_of_any_surface_tolerance` (see 4.2.747).

The data associated with a `Profile_related_positional_boundary_definition` are the following:

- `boundary_offset`;
- `defined_positional_boundary`.

4.2.751.1 boundary_offset

The `boundary_offset` specifies the `Positional_boundary_offset` (see 4.2.703) that characterizes the offset of the `Profile_related_positional_boundary` (see 4.2.750) for the `Profile_related_positional_boundary_definition`.

4.2.751.2 defined_positional_boundary

The `defined_positional_boundary` specifies the `Profile_related_positional_boundary` (see 4.2.750) that is characterized by the `Profile_related_positional_boundary_definition`.

Constraints:

WR1: The Profile_related_positional_boundary specified as the offset_positional_boundary by the Positional_boundary_and_profile_boundary_member_relationship that is specified as the offset_boundary_relationship by the Positional_boundary_offset that is specified as the boundary_offset shall be the same instance of Profile_related_positional_boundary as specified as the defined_positional_boundary.

WR2: The Profile_related_positional_boundary_definition shall be specified by exactly one Geometric_tolerance as the tolerance_zone_or_boundary_definition, and that Geometric_tolerance shall be a Boundary_based_position_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY profile_related_positional_boundary_definition
  SUBTYPE OF (positional_boundary_definition);
  SELF\positional_boundary_definition.defined_positional_boundary :
profile_related_positional_boundary;
  boundary_offset :
positional_boundary_offset;
END_ENTITY; -- profile_related_positional_boundary_definition

( *
```

4.2.752 Projected_tolerance_zone_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Projected_tolerance_zone_definition is a type of Tolerance_zone_definition (see 4.2.920) in which the Tolerance_zone (see 4.2.913) is projected from a Physical_feature_or_part_template (see 4.2.658) of the Physical_unit (see 4.2.664).

The data associated with a Projected_tolerance_zone_definition are the following:

— projected_zone_height.

4.2.752.1 projected_zone_height

The projected_zone_height specifies the Projected_zone_height_characteristic (see 4.2.754) that defines the height that the defined Tolerance_zone (see 4.2.913) is projected above a base Physical_feature_or_part_template (see 4.2.658).

Constraints:

WR1: The Tolerance_zone specified as the projected_zone of the Projected_zone_and_base_relationship that is specified as the defined_height_of of the projected_zone_height that is specified as the Projected_zone_height shall be the same instance of Projected_tolerance_zone as specified as the defined_tolerance_zone.

NOTE The attribute defined_tolerance_zone is declared in the Tolerance_zone_definition entity of which this entity is a subtype.

WR2: The Projected_tolerance_zone_definition shall be specified as the tolerance_zone_or_boundary_definition by exactly one Geometric_tolerance, and that Geometric_tolerance shall not be a Circularity_tolerance, a Cylindricity_tolerance, a Flatness_tolerance, a Straightness_tolerance, a Circular_runout_tolerance, a Total_runout_tolerance, a Profile_of_any_line_tolerance, a Profile_of_any_surface_tolerance, or a Boundary_based_position_tolerance.

WR3: The Projected_tolerance_zone_definition shall be specified as the tolerance_zone_or_boundary_definition by exactly one Geometric_tolerance, and that Geometric_tolerance shall specify either a Centre_axis, or Centre_plane as its toleranced_shape_element.

NOTE EXPRESS specification:

```

*)
ENTITY projected_tolerance_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
    projected_zone_height : projected_zone_height_characteristic;
END_ENTITY; -- projected_tolerance_zone_definition

( *
```

4.2.753 Projected_zone_and_base_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Projected_zone_and_base_relationship is a type of Managed_design_object (see 4.2.524). A Projected_zone_and_base_relationship is a relationship that associates a Tolerance_zone (see 4.2.913) with the Physical_feature_or_part_template (see 4.2.658) from which the Tolerance_zone is projected.

The data associated with a Projected_zone_and_base_relationship are the following:

- base_physical_feature;
- projected_zone;
- size_characteristic.

4.2.753.1 base_physical_feature

The base_physical_feature specifies the Physical_feature_or_part_template (see 4.2.658) from which the Tolerance_zone (see 4.2.913) that is specified as the projected_zone is projected.

4.2.753.2 projected_zone

The projected_zone specifies the Tolerance_zone (see 4.2.913) that is projected from the Physical_feature_or_part_template (see 4.2.658) that is specified as the base_physical_feature.

4.2.753.3 size_characteristic

The size_characteristic is an inverse relationship that specifies that the existence of the Projected_zone_and_base_relationship is dependent on the existence of the Projected_zone_height_characteristic (see 4.2.754) that specifies the Projected_zone_and_base_relationship as its defined_height_of.

Constraints:

IP1: The Physical_feature_or_part_template specified as the base_physical_feature shall be a feature from which it is sensible to project a Tolerance_zone.

NOTE EXPRESS specification:

```

*)
ENTITY projected_zone_and_base_relationship
  SUBTYPE OF (managed_design_object);
    base_physical_feature : physical_feature_or_part_template;
    projected_zone        : tolerance_zone;
  INVERSE
    size_characteristic : projected_zone_height_characteristic FOR
                        defined_height_of;
END_ENTITY; -- projected_zone_and_base_relationship

( *
```

4.2.754 Projected_zone_height_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Projected_zone_height_characteristic is a type of Managed_design_object (see 4.2.524). A Projected_zone_height_characteristic is a definition that characterizes the height that a Tolerance_zone (see 4.2.913) is projected from a Physical_feature_or_part_template (see 4.2.658).

The data associated with a `Projected_zone_height_characteristic` are the following:

- `defined_height_of`;
- `projected_height_value`;
- `zone_definition`.

4.2.754.1 `defined_height_of`

The `defined_height_of` specifies the `Projected_zone_and_base_relationship` (see 4.2.753) that associates a `Tolerance_zone` (see 4.2.913) with the `Physical_feature_or_part_template` (see 4.2.658) from which the `Tolerance_zone` is projected. The specified `Projected_zone_and_base_relationship` is characterized by the `Projected_zone_height_characteristic`.

4.2.754.2 `projected_height_value`

The `projected_height_value` specifies the `Length_data_element` (see 4.2.511) that represents the height that the associated `Tolerance_zone` (see 4.2.913) is projected relative to the associated `Physical_feature_or_part_template` (see 4.2.658) for the `Projected_zone_height_characteristic`.

4.2.754.3 `zone_definition`

The `zone_definition` is an inverse relationship that specifies that the existence of the `Projected_zone_height_characteristic` is dependent on the existence of the `Projected_tolerance_zone_definition` (see 4.2.752) that specifies the `Projected_zone_height_characteristic` as its `projected_zone_height`.

NOTE EXPRESS specification:

```

*)
ENTITY projected_zone_height_characteristic
  SUBTYPE OF (managed_design_object);
    defined_height_of      : projected_zone_and_base_relationship;
    projected_height_value : length_data_element;
  INVERSE
    zone_definition : projected_tolerance_zone_definition FOR
                      projected_zone_height;
END_ENTITY; -- projected_zone_height_characteristic

( *
```

4.2.755 Promissory_usage_in_product_model_relationship

A Promissory_usage_in_product_model_relationship is a type of Managed_design_object (see 4.2.524). The Promissory_usage_in_product_model_relationship is the relationship of an item to a higher assembly where the part number of the higher level assembly has not been defined.

EXAMPLE 1 A designer is designing an MCM that will be used on a board that will be part of a system. The designer does not yet know the part number for the system or that information is held in another business system. The designer does, however, know the equipment type of the system in which the MCM will be used. Using the Promissory_usage_in_product_model_relationship, the designer can specify that the MCM is promised to be used within the system, without reference to the part number of the system.

The data associated with a Promissory_usage_in_product_model_relationship are the following:

- assembly;
- component;
- security_code.

4.2.755.1 assembly

The product_model specifies the role of the Ee_product_model (see 4.2.325) for the Promissory_usage_in_product_model_relationship.

4.2.755.2 component

The component specifies the role of the Ee_product_definition (see 4.2.323) for the Promissory_usage_in_product_model_relationship.

4.2.755.3 security_code

The security_code specifies the role of the Security_classification (see 4.2.804) for the Promissory_usage_in_product_model_relationship. The security_code is optional and need not be supplied for a Promissory_usage_in_product_model_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY promissory_usage_in_product_model_relationship
  SUBTYPE OF (managed_design_object);
    component      : ee_product_definition;
    security_code   : OPTIONAL security_classification;
    product_model   : ee_product_model;
END_ENTITY; -- promissory_usage_in_product_model_relationship
( *
```

4.2.756 Promissory_usage_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Promissory_usage_relationship is a type of Managed_design_object (see 4.2.524). The Promissory_usage_relationship is the relationship of an item to a higher assembly where the next higher assembly of the item has not been defined.

EXAMPLE 1 A designer is designing an MCM that will be used on a board that will be part of a system. The designer does not yet know the next higher assembly for the MCM, since the board on which it will be placed has not yet been defined in the product structure for the system. The designer does, however, know the system in which the MCM will be used. Using the Promissory_usage_relationship, the designer can specify that the MCM is promised to be used within the system, without reference to the specific board on which it will reside.

The data associated with a Promissory_usage_relationship are the following:

- assembly;
- component;
- security_code.

4.2.756.1 assembly

The assembly specifies the role of the Physical_unit_design_view (see 4.2.667) for the Promissory_usage_relationship.

4.2.756.2 component

The component specifies the role of the Physical_unit (see 4.2.664) for the Promissory_usage_relationship.

4.2.756.3 security_code

The security_code specifies the role of the Security_classification (see 4.2.804) for the Promissory_usage_relationship. The security_code is optional and need not be supplied for a Promissory_usage_relationship.

Constraints:

UR1: The combination of the component and the assembly shall be unique within a population of Promissory_usage_relationship.

NOTE EXPRESS specification:

```
*)
ENTITY promissory_usage_relationship
  SUBTYPE OF (managed_design_object);
  component      : physical_unit;
  security_code  : OPTIONAL security_classification;
  assembly       : physical_unit_design_view;
  UNIQUE
    UR1 : component, assembly;
END_ENTITY; -- promissory_usage_relationship
( *
```

4.2.757 Purpose_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Purpose_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that specifies the intended behaviour and usage of the aspect of a design for which the requirement is allocated.

NOTE EXPRESS specification:

```
*)
ENTITY purpose_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- purpose_occurrence
( *
```

4.2.758 Qualitative_uncertainty

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Qualitative_uncertainty is a type of Uncertainty_qualifier (see 4.2.936). The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY qualitative_uncertainty
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : text;
END_ENTITY;
( *
```

4.2.759 Radius_edge_feature_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Radius_edge_feature_shape is a type of Edge_shape (see 4.2.313) that describes a circular cross sectioned edge. See Bevel_edge_feature_shape (see 4.2.69) for the additional required geometric information that shall be provided.

EXAMPLE 1 Figure 100 illustrates the shape related information exchanged with this Application object.

The data associated with a Radius_edge_feature_shape are the following:

- maximum_radius_length;
- minimum_radius_length.

4.2.759.1 maximum_radius_length

The maximum_radius_length specifies the Length_data_element (see 4.2.511) of the largest allowable radius of the circle that is tangent to the edge surface and to the lateral surface on which the edge surface is a feature for the Radius_edge_feature_shape. The included angle between the points of tangency shall always be 90 degrees.

4.2.759.2 minimum_radius_length

The minimum_radius_length specifies the Length_data_element (see 4.2.511) of the smallest allowable radius of the circle that is tangent to the edge surface and to the lateral surface on which the edge surface is a feature for the Radius_edge_feature_shape. The included angle between the points of tangency shall always be 90 degrees.

NOTE EXPRESS specification:

```
* )
ENTITY radius_edge_feature_shape
  SUBTYPE OF (edge_shape);
    minimum_radius_length : length_data_element;
    maximum_radius_length : length_data_element;
END_ENTITY; -- radius_edge_feature_shape

--tt tc1 wd8 correction
(*
```

4.2.760 Range_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

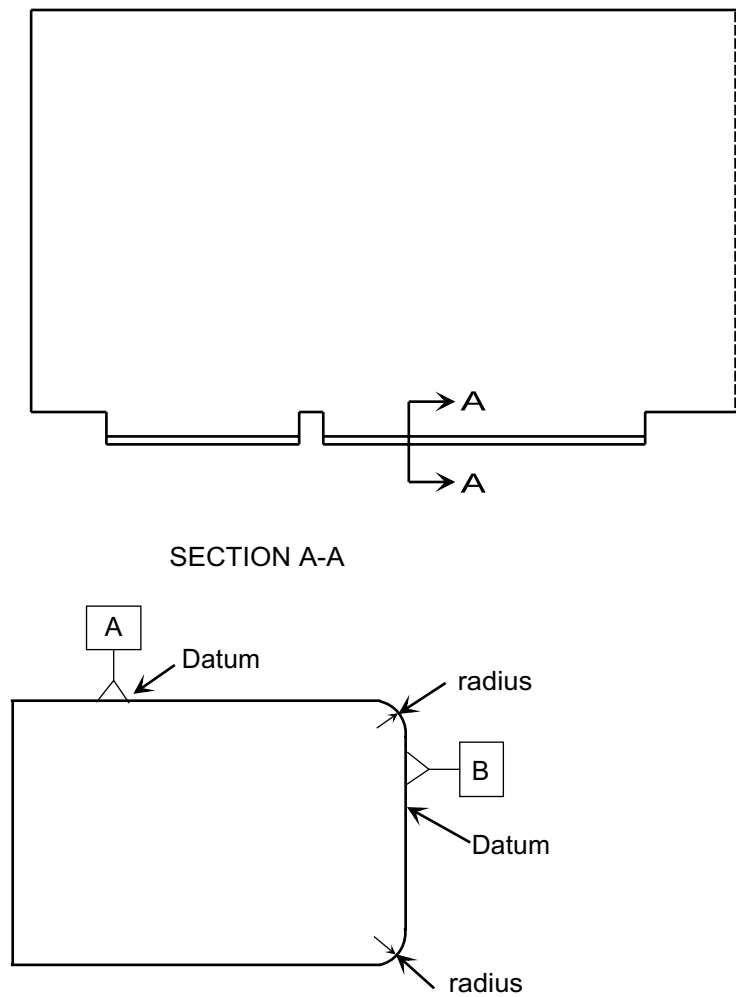


Figure 100 – Radius_edge_feature_shape

A Range_characteristic is a type of Coordinated_characteristic (see 4.2.197) that specifies a range of values. The range may be open or closed. The range may be single ended.

The data associated with a Range_characteristic are the following:

- component_characteristic;
- range_type.

4.2.760.1 component_characteristic

The component_characteristic specifies the value of the Numeric_parameter (see 4.2.565) for the Range_characteristic. There shall be exactly two Numeric_parameter for the Range_characteristic.

4.2.760.2 range_type

The range_type specifies the classification of the Range_characteristic. The Range_characteristic shall be open, closed, lower_open or upper_open. Open shall be interpreted as not including the end point or points in the range. Closed shall be interpreted as including the end point or points in the range.

NOTE EXPRESS specification:

```

*)
ENTITY range_characteristic
  SUBTYPE OF (coordinated_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET [2:2] OF
                                                                numeric_parameter;
    range_type: range_class;
END_ENTITY; -- range_characteristic
( *
```

4.2.761 Ratio_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Ratio_data_element is a

The data associated with a Ratio_data_element are the following:

— additional_data.

4.2.761.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```

*)
ENTITY ratio_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- ratio_data_element
( *
```

4.2.762 Rectangular_area

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rectangular_area is a type of Primitive_2d (see 4.2.720).

The data associated with a Rectangular_area are the following:

- position;
- x;
- y.

4.2.762.1 position

The position specifies an Axis_placement_2d (see 4.2.57) for the Rectangular_area.

4.2.762.2 x

The x specifies a Length_data_element (see 4.2.511) for the Rectangular_area.

4.2.762.3 y

The y specifies a Length_data_element (see 4.2.511) for the Rectangular_area.

NOTE EXPRESS specification:

```
*)
ENTITY rectangular_area
  SUBTYPE OF (primitive_2d);
  position: axis_placement_2d;
  x: length_data_element;
  y: length_data_element;
END_ENTITY;
```

(*

4.2.763 Rectangular_array_placement_group_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rectangular_array_placement_group_component is a type of Array_placement_group (see 4.2.29). The location_uncertainty refers to originating CAD system geometric uncertainty, not a material processing tolerance.

The data associated with a `Rectangular_array_placement_group_component` are the following:

- `element`;
- `initial_row_positive_y_displacement`;
- `terminus_row_positive_y_displacement`.

4.2.763.1 `element`

The `element` specifies the role of a `Linear_array_placement_group_component_link` (see 4.2.517) for the `Rectangular_array_placement_group_component`. There shall be one or more `Linear_array_placement_group_component_link` for a `Rectangular_array_placement_group`.

4.2.763.2 `initial_row_positive_y_displacement`

The `initial_row_positive_y_displacement` specifies a `Length_data_element` (see 4.2.511) for the `Rectangular_array_placement_group_component`.

4.2.763.3 `terminus_row_positive_y_displacement`

The `terminus_row_positive_y_displacement` specifies a `Length_data_element` (see 4.2.511) for the `Rectangular_array_placement_group_component`.

NOTE EXPRESS specification:

```

*)
ENTITY rectangular_array_placement_group_component
  SUBTYPE OF (array_placement_group);
  element : SET [1:?] OF linear_array_placement_group_component_link;
  initial_row_positive_y_displacement : length_data_element;
  terminus_row_positive_y_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- rectangular_array_placement_group_component
( *
```

4.2.764 `Reference_composition_path`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Reference_composition_path` is a type of `Managed_design_object` (see 4.2.524). A `Reference_composition_path` is one path through the function reference network graph.

The data associated with a Reference_composition_path are the following:

- path_element;
- reference_functional_unit.

4.2.764.1 path_element

The path_element specifies the role of the Functional_unit (see 4.2.400) for the Reference_composition_path. There shall be one or more Functional_unit for a Reference_composition_path.

4.2.764.2 reference_functional_unit

The reference_functional_unit specifies the Functional_unit (see 4.2.400) that participates as a link to the leaf node in the functional decomposition hierarchical definition in the Reference_composition_path.

NOTE EXPRESS specification:

```

*)
ENTITY reference_composition_path
  SUBTYPE OF (managed_design_object);
    path_element          : SET [1:?] OF functional_unit;
    reference_functional_unit : functional_unit;
END_ENTITY; -- reference_composition_path

( *
```

4.2.765 Reference_functional_unit_assignment_to_part

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Reference_functional_unit_assignment_to_part is a type of Managed_design_object (see 4.2.524). A Reference_functional_unit_assignment_to_part assigns one specific path in the function reference network graph to an implementation. A Reference_functional_unit_assignment_to_part provides support for exchange of information needed to determine swappability of functions assigned to devices.

The data associated with a `Reference_functional_unit_assignment_to_part` are the following:

- `functional_design_definition_path`;
- `implementation`;
- `of_property`;
- `path_alias`;
- `swap_code`.

4.2.765.1 functional_design_definition_path

The `functional_design_definition_path` specifies the role of the `Design_composition_path` (see 4.2.262) for the `Reference_functional_unit_assignment_to_part`.

4.2.765.2 implementation

The `implementation` specifies the role of the `Physical_unit_usage_view` (see 4.2.672) for the `Reference_functional_unit_assignment_to_part`.

4.2.765.3 of_property

The `of_property` specifies the role of the `Coordinated_characteristic` for the `Reference_functional_unit_assignment_to_part`. There may be more than one `of_property` for a `Reference_functional_unit_assignment_to_part`. The `of_property` need not be specified for a particular `Reference_functional_unit_assignment_to_part`.

4.2.765.4 path_alias

The `path_alias` specifies the string that is the identifier for the `Reference_functional_unit_assignment_to_part`.

4.2.765.5 swap_code

The `swap_code` specifies the string that is the type of exchange that is valid.

Constraints:

UR1: The combination of the `implementation` and the `path_alias` shall be unique within a population of `Reference_functional_unit_assignment_to_part`.

NOTE EXPRESS specification:

```

*)
ENTITY reference_functional_unit_assignment_to_part
  SUBTYPE OF (managed_design_object);
    functional_design_definition_path : reference_composition_path;
    implementation                     : physical_unit_or_part_template;
    of_property                       : OPTIONAL SET [1:?] OF
                                      coordinated_characteristic;

    path_alias                        : ee_name;
    swap_code                         : OPTIONAL STRING;
  UNIQUE
    UR1 : implementation, path_alias;
END_ENTITY; -- reference_functional_unit_assignment_to_part

--tt tc1 wd9 added
(*)

```

4.2.766 Reference_packaged_part_assembly_implementation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Reference_packaged_part_assembly_implementation is a type of Assembly_module (see 4.2.43).

NOTE EXPRESS specification:

```

*)
ENTITY reference_packaged_part_assembly_implementation
  SUBTYPE OF (assembly_module);
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*
    USEDIN(SELF\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | NOT
      (
        SIZEOF (QUERY ( epcr &#60;* USEDIN(prpc,
          'AP210_ARM.EE_PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY') |
          (epcr.sub_category\ee_product_category.name =
            'reference packaged part assembly model')
        )) = 1
      )
    )) = 0;
END_ENTITY; -- reference_packaged_part_assembly_implementation

--tt tc1 wd9 added
(*)

```

4.2.767 Reference_packaged_part_interconnect_implementation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Reference_packaged_part_interconnect_implementation is a type of Interconnect_module (see 4.2.440).

NOTE EXPRESS specification:

```

*)
ENTITY reference_packaged_part_interconnect_implementation
  SUBTYPE OF (interconnect_module);
WHERE
  WR1: SIZEOF (QUERY( prpc &#60;*
    USEDIN(SELF\ee_product_definition.version.versioned_item,
      'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | NOT
      (
        SIZEOF (QUERY ( epcr &#60;* USEDIN(prpc,
          'AP210_ARM.EE_PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY') |
          (epcr.sub_category\ee_product_category.name =
            'reference packaged part interconnect model')
        )) = 1
      )
    )) = 0;
END_ENTITY; -- reference_packaged_part_interconnect_implementation

( *
```

4.2.768 Registered_font

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Registered_font is a type of Managed_design_object (see 4.2.524). A Registered_font is a font that is registered by an organization and defined within a document. Support shall be provided for font character representation using outline glyphs and font family exchange.

The data associated with a Registered_font are the following:

- font_code;
- registration_document.

4.2.768.1 font_code

The font_code specifies the string for the identifier of the Registered_font. The font_code shall be an entry in the registration_document.

4.2.768.2 registration_document

The registration_document specifies the Ee_document (see 4.2.316) that contains the information about the Registered_font.

Constraints:

UR1: The combination of the font_code and the registration_document shall be unique within all members of a population of Registered_font.

NOTE EXPRESS specification:

```
*)
ENTITY registered_font
  SUBTYPE OF (managed_design_object);
    registration_document : ee_document;
    font_code             : STRING;
  UNIQUE
    UR1 : font_code, registration_document;
END_ENTITY; -- registered_font

( *
```

4.2.769 Requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Requirement_allocation is a type of Managed_design_object (see 4.2.524). A Requirement_allocation is the allocation of requirement occurrences to the items in the product that meet the requirements.

The data associated with a Requirement_allocation are the following:

- allocated_requirement;
- target_product_object.

4.2.769.1 allocated_requirement

The allocated_requirement specifies the Ee_requirement_occurrence (see 4.2.330) that is being allocated in the Requirement_allocation.

4.2.769.2 target_product_object

The `target_product_object` specifies either an `Ee_product` (see 4.2.319), an `Ee_product_configuration` (see 4.2.322), an `Ee_product_definition` (see 4.2.323), an `Ee_product_version` (see 4.2.328), a `Part_feature` (see 4.2.609), a `Physical_connectivity_definition` (see 4.2.652), or a `Physical_connectivity_element` (see 4.2.654), or one of an `Assembly_component` (see 4.2.34), a `Component_external_reference` (see 4.2.117), a `Component_feature` (see 4.2.118), a `Component_feature_external_reference` (see 4.2.119), an `Ee_requirement_occurrence` (see 4.2.330), a `Stratum` (see 4.2.845), or one of an `Inter_stratum_feature` (see 4.2.432), a `Laminate_component` (see 4.2.483), a `Stratum_feature` (see 4.2.851), or a `Stratum_surface` (see 4.2.868) as the target for the `Requirement_allocation`. There shall be one or more `Ee_product`, `Ee_product_configuration`, `Ee_product_definition`, `Ee_product_version`, `Part_feature`, `Physical_connectivity_definition`, `Assembly_component`, `Component_external_reference`, `Component_feature`, `Component_feature_external_reference`, `Ee_requirement_occurrence`, `Stratum`, `Inter_stratum_feature`, `Laminate_component`, `Stratum_feature`, or `Stratum_surface` for the `Requirement_allocation`.

NOTE EXPRESS specification:

```
*)
ENTITY requirement_allocation
  SUBTYPE OF (managed_design_object);
    target_product_object : SET [1:?] OF product_object_select;
    allocated_requirement : ee_requirement_occurrence;
END_ENTITY; -- requirement_allocation

(*
```

4.2.770 Requirement_composition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Requirement_composition` is a type of `Managed_design_object` (see 4.2.524). A `Requirement_composition` helps to compose an aggregation of `Ee_requirement_occurrence` (see 4.2.330) into a group in order to construct a requirements structure.

EXAMPLE 1 A requirements table is a typical example of a requirements structure.

EXAMPLE 2 A `Requirement_composition` would be used to convey an aggregate requirement that is the result of combining two requirements.

The data associated with a `Requirement_composition` are the following:

- composition;
- element.

4.2.770.1 composition

The composition specifies the Requirement_group_occurrence (see 4.2.772) for the aggregation for the Requirement_composition.

4.2.770.2 element

The element specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Requirement_composition. There shall be one or more Ee_requirement_occurrence for a Requirement_composition.

NOTE EXPRESS specification:

```

*)
ENTITY requirement_composition
  SUBTYPE OF (managed_design_object);
    element      : SET [1:?] OF ee_requirement_occurrence;
    composition  : requirement_group_occurrence;
END_ENTITY; -- requirement_composition

--tt tc1 wd9 added
( *
```

4.2.771 Requirement_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Requirement_definition is a type of Ee_product_definition (see 4.2.323).

The data associated with a Requirement_definition are the following:

— associated_requirement_occurrence.

4.2.771.1 associated_requirement_occurrence

The associated_requirement_occurrence is an inverse relationship that specifies that the existence of the Requirement_definition is dependent on the existence of a Ee_requirement_occurrence to reference the Requirement_definition.

NOTE EXPRESS specification:

```

*)
ENTITY requirement_definition
  SUBTYPE OF (ee_product_definition);
INVERSE
  associated_requirement_occurrence :
    ee_requirement_occurrence FOR associated_definition;
END_ENTITY; -- requirement_definition
( *

```

4.2.772 Requirement_group_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Requirement_group_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that is a collection of requirements, aggregated by providing one or more Requirement_composition.

NOTE EXPRESS specification:

```

*)
ENTITY requirement_group_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- requirement_group_occurrence
( *

```

4.2.773 Resistance_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Resistance_data_element is a

The data associated with a Resistance_data_element are the following:

— additional_data.

4.2.773.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY resistance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- resistance_data_element

( *
```

4.2.774 Restraint_condition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Restraint_condition` is a type of `Managed_design_object` (see 4.2.524). A `Restraint_condition` is the specification of a requirement on the restraining conditions applied either to a `Datum_feature` (see 4.2.226) or a `Physical_feature_or_part_template` (see 4.2.658) during the verification of one or more `Geometric_tolerance` (see 4.2.419) or `Dimensional_characteristic` (see 4.2.298). The `Restraint_condition` may be a `Tolerance_specific_restraint_condition` (see 4.2.912). If the `Restraint_condition` is not a `Tolerance_specific_restraint_condition`, the `Restraint_condition` is a requirement for all instances of `Geometric_tolerance` and `Dimensional_characteristic` with the exception of those instances of `Geometric_tolerance` that have a `Pre_defined_geometric_tolerance_qualifier` (see 4.2.708) applied to them that have a description of "FREE_STATE", and those instances of `Dimensional_characteristic` that have a `Pre_defined_dimension_qualifier` (see 4.2.707) applied to them that have a description of "FREE_STATE".

The data associated with a `Restraint_condition` are the following:

- `applied_to`;
- `restraint_description`.

4.2.774.1 applied_to

The `applied_to` specifies either a `Physical_feature_or_part_template` (see 4.2.658) or a `Datum_feature_usage_in_datum_relationship` (see 4.2.229). If the `applied_to` specifies a `Physical_feature_or_part_template`, it is on that `Physical_feature_or_part_template` that the restraint requirement is applied. If a `Datum_feature_usage_in_datum_relationship` is specified as the `applied_to`, the `Restraint_condition` specifies a restraint requirement on the relationship between the `Datum_feature` (see 4.2.226) and the `Datum` (see 4.2.214) specified by the `Datum_feature_usage_in_datum_relationship`.

4.2.774.2 restraint_description

The `restraint_description` specifies the `Ee_requirement_occurrence` (see 4.2.330) that describes how the restraint shall be applied.

EXAMPLE 1 The `Restraint_condition` might specify the torque to apply to bolts to secure `Datum_feature` (see 4.2.226)"A" to inspection equipment that simulates `Datum` (see 4.2.214)"A".

NOTE EXPRESS specification:

```
*)
ENTITY restraint_condition
  SUPERTYPE OF (tolerance_specific_restraint_condition)
  SUBTYPE OF (managed_design_object);
  restraint_description : ee_requirement_occurrence;
  applied_to            : restrained_item_select;
END_ENTITY; -- restraint_condition
```

(*

4.2.775 Right_angular_wedge

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Right_angular_wedge is a type of Csg_primitive (see 4.2.201) which is used in forming a "CSG" solid model.

NOTE EXPRESS specification:

```
*)
ENTITY right_angular_wedge
  SUBTYPE OF (csg_primitive);
END_ENTITY; -- right_angular_wedge
```

(*

4.2.776 Right_circular_cylinder

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Right_circular_cylinder is a type of Csg_primitive (see 4.2.201) which is used in forming a "CSG" solid model.

NOTE EXPRESS specification:

```
*)
ENTITY right_circular_cylinder
  SUBTYPE OF (csg_primitive);
END_ENTITY; -- right_circular_cylinder
```

(*

4.2.777 Routed_interconnect_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Routed_interconnect_component is a type of Assembly_component (see 4.2.34). The functionality of a Routed_interconnect_component shall be that of a short.

The data associated with a Routed_interconnect_component are the following:

— routed_shape.

EXAMPLE 1 An example of a Routed_interconnect_component is a jumper wire. A jumper wire does not have a predefined placement or shape for its terminal. The shape of a jumper wire is not predefined. The predefined shapes in this example would be those shapes defined in a library.

4.2.777.1 routed_shape

The routed_shape specifies the role of the Feature_shape_definition (see 4.2.372) for the Routed_interconnect_component.

Constraints:

WR1: All terminals defined for the Routed_interconnect_component shall be equipotential with respect to the same reference potential.

NOTE EXPRESS specification:

```
* )
ENTITY routed_interconnect_component
--tt tc1 wd11 modified
  SUBTYPE OF (physical_component);
--tt tc1 wd9 modified
  routed_centrelines_shape : curve;
END_ENTITY; -- routed_interconnect_component

--tt tc1 wd12 modified name
(*
```

4.2.778 Routed_physical_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Routed_physical_component is a

The data associated with a Routed_physical_component are the following:

— additional_data.

4.2.778.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY routed_physical_component
  SUBTYPE OF (physical_component);
--tt tc1 wd9 modified
    routed_centreline_shape : curve;
END_ENTITY; -- routed_physical_component

--tt tc1 wd12 added
(*)
```

4.2.779 Routed_printed_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Routed_printed_component is a type of Printed_component (see 4.2.721).

The data associated with a Routed_printed_component are the following:

- component_extent;
- routed_shape;
- usage_definition.

EXAMPLE 1 An example of a Routed_printed_component is a meandering stripline.

4.2.779.1 component_extent

The component_extent specifies the role of the Inter_stratum_extent (see 4.2.431) for the Routed_printed_component.

4.2.779.2 routed_shape

The `routed_shape` specifies the role of the `Feature_shape_definition` (see 4.2.372) for the `Routed_printed_component`. The `routed_shape` may be represented as the result of an extrusion process, where the `usage_definition` is extruded along the path of the route.

NOTE This representation is not intended to reflect physical processing.

NOTE Usually there is one `stratum_feature_implementation` for each `Design_layer_stratum` used in the implementation.

4.2.779.3 usage_definition

The `usage_definition` specifies the role of the `Printed_part_cross_section_template` (see 4.2.728) for the `Routed_printed_component`. The `usage_definition` provides a cross-sectional definition that shall be interpreted as a constant cross-section along the path of the route.

NOTE EXPRESS specification:

```
*)
ENTITY routed_printed_component
  SUBTYPE OF (printed_component);
    routed_centrelines_shape : curve;
    SELF\assembly_component.usage_definition : printed_part_cross_section_template;
    component_extent          : inter_stratum_extent;
END_ENTITY; -- routed_printed_component
(*
```

4.2.780 Rule_action

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Rule_action` is a type of `Managed_design_object` (see 4.2.524). A `Rule_action` is one of a `Rule_adoption` (see 4.2.781), a `Rule_change_request` (see 4.2.784), a `Rule_creation` (see 4.2.785), a `Rule_expiration` (see 4.2.786), a `Rule_modification` (see 4.2.788), a `Rule_rejection` (see 4.2.790), a `Rule_request` (see 4.2.791), a `Rule_submission` (see 4.2.794), or a `Rule_supersedence` (see 4.2.795). A `Rule_action` is an assertion that some action has occurred toward management of rules.

The data associated with a `Rule_action` are the following:

- `subject_action_assignment`;
- `subject_rule`.

4.2.780.1 subject_action_assignment

The subject_action_assignment specifies the role of the Participant_date_and_time_assignment (see 4.2.637) for the Rule_action.

4.2.780.2 subject_rule

The subject_rule specifies the Ee_rule_version (see 4.2.337) as the version of an Ee_rule (see ??) for which some management Rule_action has occurred.

Constraints:

UR1: The combination of the subject_action_assignment and the subject_rule shall be unique within a population of Rule_actions.

NOTE EXPRESS specification:

```

*)
ENTITY rule_action
  ABSTRACT SUPERTYPE OF (ONEOF (rule_submission,rule_adoption,
    rule_rejection,rule_supersedence,rule_creation,rule_expiration,
    rule_change_request,rule_request,rule_modification))
  SUBTYPE OF (managed_design_object);
  subject_action_assignment : participant_date_and_time_assignment;
  subject_rule               : ee_rule_version;
  UNIQUE
    UR1 : subject_rule, subject_action_assignment;
END_ENTITY; -- rule_action

```

(*

4.2.781 Rule_adoption

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_adoption is a type of Rule_action (see 4.2.780) that states a version of an Ee_rule (see ??) has been adopted.

NOTE EXPRESS specification:

```
* )  
ENTITY rule_adoption  
  SUBTYPE OF (rule_action);  
END_ENTITY; -- rule_adoption  
  
( *
```

4.2.782 Rule_based_constraint_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_based_constraint_occurrence is a type of Constraint_occurrence (see 4.2.193) that limits possible solutions to the requirements specification by identifying boundary conditions for any potential solution. A Rule_based_constraint_occurrence specifies a rule that must be satisfied.

EXAMPLE 1 A rule might exist that requires junction temperatures of Silicon based integrated devices to be less than 105 degrees Centrigade.

The data associated with a Rule_based_constraint_occurrence are the following:

— design_constraint.

4.2.782.1 design_constraint

The design_constraint specifies the role of the Ee_rule_version (see 4.2.337) for the Rule_based_constraint_occurrence.

NOTE EXPRESS specification:

```
* )  
ENTITY rule_based_constraint_occurrence  
  SUBTYPE OF (constraint_occurrence);  
    design_constraint : ee_rule_version;  
END_ENTITY; -- rule_based_constraint_occurrence  
  
( *
```

4.2.783 Rule_based_evaluated_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_based_evaluated_characteristic is a type of Evaluated_characteristic (see 4.2.356). The Rule_based_evaluated_characteristic identifies the Ee_rule_version (see 4.2.337) used to validate or evaluate a Planned_characteristic (see 4.2.679).

The data associated with a Rule_based_evaluated_characteristic are the following:

— evaluating_rule.

4.2.783.1 evaluating_rule

The evaluating_rule specifies the role of the Ee_rule (see ??) for a Rule_based_evaluated_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY rule_based_evaluated_characteristic
  SUBTYPE OF (evaluated_characteristic);
    evaluating_rule : ee_rule_version;
END_ENTITY; -- rule_based_evaluated_characteristic
(*
```

4.2.784 Rule_change_request

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_change_request is a type of Rule_action (see 4.2.780) that states a request has been made for modification to a version of an Ee_rule (see ??). The change reason shall be provided as a part of the Rule_change_request.

The data associated with a Rule_change_request are the following:

— change_reason.

4.2.784.1 change_reason

The change_reason specifies the human-interpretable reason for requesting a Rule_change_request.

NOTE EXPRESS specification:

```
*)
ENTITY rule_change_request
  SUBTYPE OF (rule_action);
    change_reason : STRING;
END_ENTITY; -- rule_change_request
(*
```

4.2.785 Rule_creation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_creation is a type of Rule_action (see 4.2.780) that states that a version of an Ee_rule (see ??) has been created.

NOTE EXPRESS specification:

```
*)  
ENTITY rule_creation  
    SUBTYPE OF (rule_action);  
END_ENTITY; -- rule_creation
```

(*

4.2.786 Rule_expiration

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_expiration is a type of Rule_action (see 4.2.780) that states that a version of an Ee_rule (see ??) has expired.

The data associated with a Rule_expiration are the following:

— expiration_rationale.

4.2.786.1 expiration_rationale

The expiration_rationale specifies the human-interpretable reason for the Rule_expiration.

NOTE EXPRESS specification:

```
*)  
ENTITY rule_expiration  
    SUBTYPE OF (rule_action);  
    expiration_rationale : STRING;  
END_ENTITY; -- rule_expiration
```

(*

4.2.787 Rule_justification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_justification is a type of Managed_design_object (see 4.2.524). A Rule_justification is the justification for an Ee_rule (see ??).

The data associated with a Rule_justification are the following:

- justification_rationale;
- justified_action.

4.2.787.1 justification_rationale

The justification_rationale specifies the text that describes the Rule_justification.

4.2.787.2 justified_action

The justified_action specifies the Rule_action (see 4.2.780) that results from the Rule_justification.

NOTE EXPRESS specification:

```
*)
ENTITY rule_justification
  SUBTYPE OF (managed_design_object);
    justified_action      : rule_action;
    justification_rationale : STRING;
END_ENTITY; -- rule_justification
```

(*

4.2.788 Rule_modification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_modification is a type of Rule_action (see 4.2.780) that states a version of an Ee_rule (see ??) is a result of a request to change an existing version of an Ee_rule.

The data associated with a Rule_modification are the following:

- modification_rationale.

4.2.788.1 modification_rationale

The modification_rationale specifies the role of the Rule_change_request (see 4.2.784) for the Rule_modification.

NOTE EXPRESS specification:

```
*)  
ENTITY rule_modification  
  SUBTYPE OF (rule_action);  
    modification_rationale : rule_change_request;  
END_ENTITY; -- rule_modification  
  
(*
```

4.2.789 Rule_priority

A Rule_priority is a type of Managed_design_object (see 4.2.524). A Rule_priority is a statement that an Ee_product (see 4.2.319) that is in the rule category has a priority value. This value is used to order rules. The priority is applied to the rule through the Ee_rule_definition. It is not the Ee_rule_definition that has the priority.

The data associated with a Rule_priority are the following:

- prioritized_rule;
- priority.

4.2.789.1 prioritized_rule

The prioritized_rule specifies the role of the Ee_rule_definition (see 4.2.333) for the Rule_priority.

4.2.789.2 priority

The priority specifies the integer for the priority number of the Rule_priority. The lowest numeric value shall have the highest priority.

Constraints:

WR1: A number less than zero shall not be allowed.

NOTE EXPRESS specification:

```

*)
ENTITY rule_priority
  SUBTYPE OF (managed_design_object);
  priority      : INTEGER;
  prioritized_rule : ee_rule_definition;
WHERE
  WR1: priority &#62;= 0;
END_ENTITY; -- rule_priority
( *

```

4.2.790 Rule_rejection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_rejection is a type of Rule_action (see 4.2.780) that states a version of an Ee_rule (see ??) has been rejected. The rejection rationale shall be part of the Rule_rejection.

The data associated with a Rule_rejection are the following:

— rejection_reason.

4.2.790.1 rejection_reason

The rejection_reason specifies the human-interpretable explanation for the Rule_rejection.

NOTE EXPRESS specification:

```

*)
ENTITY rule_rejection
  SUBTYPE OF (rule_action);
  rejection_reason : STRING;
END_ENTITY; -- rule_rejection
( *

```

4.2.791 Rule_request

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_request is a type of Rule_action (see 4.2.780) that states a version of an Ee_rule (see ??) has been requested. The rationale for the version shall be part of the Rule_request.

NOTE EXPRESS specification:

```
* )  
ENTITY rule_request  
  SUBTYPE OF (rule_action);  
END_ENTITY; -- rule_request  
  
(*
```

4.2.792 Rule_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_set is a type of Managed_design_object (see 4.2.524). A Rule_set may be a Rule_set_group (see 4.2.793). A Rule_set is an association among Ee_rule (see ??) with Rule_priority (see 4.2.789).

The data associated with a Rule_set are the following:

- identifier;
- purpose;
- rule_priority_member.

4.2.792.1 identifier

The identifier specifies the human-interpretable identification for the Rule_set.

4.2.792.2 purpose

The purpose specifies the human-interpretable string for the purpose for the Rule_set.

4.2.792.3 rule_priority_member

The rule_priority_member specifies the role of the Rule_priority (see 4.2.789) for the Rule_set. There shall be one or more Rule_priority for the Rule_set.

Constraints:

UR1: The identifier shall be unique within a population of Rule_sets.

NOTE EXPRESS specification:

```

*)
ENTITY rule_set
  SUPERTYPE OF (rule_set_group)
  SUBTYPE OF (managed_design_object);
  rule_priority_member : SET [1:?] OF rule_priority;
  identifier            : STRING;
  purpose               : STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- rule_set

```

(*

4.2.793 Rule_set_group

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_set_group is a type of Rule_set (see 4.2.792). A Rule_set_group shall have at least two Rule_sets.

The data associated with a Rule_set_group are the following:

— elements.

4.2.793.1 elements

The elements specify the Rule_set (see 4.2.792) for the Rule_set_group. There may be more than two Rule_set for the Rule_set_group.

NOTE EXPRESS specification:

```

*)
ENTITY rule_set_group
  SUBTYPE OF (rule_set);
  elements : SET [2:?] OF rule_set;
END_ENTITY; -- rule_set_group

```

(*

4.2.794 Rule_submission

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_submission is a type of Rule_action (see 4.2.780) stating that a version of an Ee_rule (see ??) has been submitted.

The data associated with a Rule_submission are the following:

— submission_rationale.

4.2.794.1 submission_rationale

The submission_rationale specifies the text that describes the rationale for the Rule_submission.

NOTE EXPRESS specification:

```
*)  
ENTITY rule_submission  
  SUBTYPE OF (rule_action);  
    submission_rationale : STRING;  
END_ENTITY; -- rule_submission  
  
(*
```

4.2.795 Rule_supersedence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Rule_supersedence is a type of Rule_action (see 4.2.780) stating that one Ee_rule_version (see 4.2.337) has been superseded by another Ee_rule_version. Both shall be versions of the same Ee_rule (see ??).

The data associated with a Rule_supersedence are the following:

— superseded_rule.

4.2.795.1 superseded_rule

The superseded_rule specifies the Ee_rule_version (see 4.2.337) that has been superseded for the Rule_supersedence.

Constraints:

WR1: The Ee_rule in the role of configuration_managed_rule for the Ee_rule_version in the role of superseded_rule specified by this Rule_supersedence shall be the same Ee_rule that is in the role of configuration_managed_rule inherited from the supertype Rule_action that is specified by this Rule_supersedence.

NOTE EXPRESS specification:

```

*)
ENTITY rule_supersedence
  SUBTYPE OF (rule_action);
  superseded_rule : ee_rule_version;
END_ENTITY; -- rule_supersedence

( *
```

4.2.796 Runout_tolerance_zone_definition_with_specified_angle

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Runout_tolerance_zone_definition_with_specified_angle is a type of Tolerance_zone_definition (see 4.2.920) that is defined by either a Circular_runout_tolerance (see 4.2.102) or a Total_runout_tolerance (see 4.2.929), and in which the Tolerance_zone (see 4.2.913) is oriented with respect to a Directed_axis (see 4.2.303). The Directed_axis is coincident with the Datum_axis (see 4.2.215) about which the Tolerance_zone is centred.

NOTE Circular run-out tolerance in a specified direction is described in 14.13.4 of ISO 1101.

The data associated with a Runout_tolerance_zone_definition_with_specified_angle are the following:

— runout_orientation.

4.2.796.1 runout_orientation

The runout_orientation specifies the Runout_zone_orientation (see 4.2.798) that defines the orientation of the Tolerance_zone (see 4.2.913) relative to a specified Directed_axis (see 4.2.303) for the Runout_tolerance_zone_definition_with_specified_angle.

Constraints:

WR1: The Tolerance_zone specified as the oriented_zone by the Runout_tolerance_zone_orienting_relationship that is specified as the oriented_relationship by the Runout_zone_orientation that is specified as the runout_orientation, shall be the same instance of Tolerance_zone as specified as the defined_tolerance_zone.

WR2: The Runout_tolerance_zone_definition_with_specified_angle shall be specified as the tolerance_zone_or_boundary_definition by exactly one Geometric_tolerance, and that Geometric_tolerance shall be either a Circular_runout_tolerance or a Total_runout_tolerance.

WR3: The instance of Centre_axis specified as the axis_to_be_oriented by the Directed_axis to which the Tolerance_zone is oriented shall be a Datum_axis that is in the Datum_system the definition of which is specified as the datum_system_definition by the associated Circular_runout_tolerance or Total_runout_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY runout_tolerance_zone_definition_with_specified_angle
  SUBTYPE OF (tolerance_zone_definition);
  runout_orientation : runout_zone_orientation;
END_ENTITY; -- runout_tolerance_zone_definition_with_specified_angle

( *

```

4.2.797 Runout_tolerance_zone_orienting_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Runout_tolerance_zone_orienting_relationship is a type of Managed_design_object (see 4.2.524). A Runout_tolerance_zone_orienting_relationship is a relationship that associates a Tolerance_zone (see 4.2.913) with the Directed_axis (see 4.2.303) with which it is oriented.

The data associated with a Runout_tolerance_zone_orienting_relationship are the following:

- angular_size_characteristic;
- oriented_zone;
- orienting_axis.

4.2.797.1 angular_size_characteristic

The angular_size_characteristic is an inverse relationship that specifies that the existence of the Runout_tolerance_zone_orienting_relationship is dependent on the existence of the Runout_zone_orientation (see 4.2.798) that specifies the Runout_tolerance_zone_orienting_relationship as its oriented_relationship.

4.2.797.2 oriented_zone

The oriented_zone specifies the Tolerance_zone (see 4.2.913) that is oriented with respect to the specified Directed_axis (see 4.2.303) by the Runout_tolerance_zone_orienting_relationship.

4.2.797.3 orienting_axis

The orienting_axis specifies the Directed_axis (see 4.2.303) to which the specified Tolerance_zone (see 4.2.913) is oriented by the Runout_tolerance_zone_orienting_relationship.

Constraints:

IP1: The Centre_axis upon which the Directed_axis is based shall correspond with the axis upon which the unit would be rotated if the corresponding run-out tolerance (i.e., either a Circular_runout_tolerance or Total_runout_tolerance) were to be verified.

NOTE EXPRESS specification:

```

*)
ENTITY runout_tolerance_zone_orienting_relationship
  SUBTYPE OF (managed_design_object);
    oriented_zone : tolerance_zone;
    orienting_axis : directed_axis;
  INVERSE
    angular_size_characteristic : runout_zone_orientation FOR oriented_relationship;
END_ENTITY; -- runout_tolerance_zone_orienting_relationship

( *
```

4.2.798 Runout_zone_orientation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Runout_zone_orientation is a type of Managed_design_object (see 4.2.524). A Runout_zone_orientation is a definition that characterizes the orientation between a Tolerance_zone (see 4.2.913) and a Directed_axis (see 4.2.303) that is coincident with the Datum_axis (see 4.2.215) about which the Tolerance_zone is centred.

The data associated with a Runout_zone_orientation are the following:

- oriented_relationship;
- semi_angle_value;
- zone_definition.

4.2.798.1 oriented_relationship

The oriented_relationship specifies the Runout_tolerance_zone_orienting_relationship (see 4.2.797) that associates the Tolerance_zone (see 4.2.913) with the Directed_axis (see 4.2.303) with which the Tolerance_zone is oriented, and that is characterized by the Runout_zone_orientation.

4.2.798.2 semi_angle_value

The semi_angle_value specifies the Angle_data_element (see 4.2.23) that represents the size of the angle between the Tolerance_zone (see 4.2.913) and the Directed_axis (see 4.2.303).

4.2.798.3 zone_definition

The zone_definition is an inverse relationship that specifies that the existence of the Runout_zone_orientation is dependent on the existence of the Runout_tolerance_zone_definition_with_specified_angle (see 4.2.796) that specifies the Runout_zone_orientation as its runout_orientation.

NOTE EXPRESS specification:

```
*)
ENTITY runout_zone_orientation
  SUBTYPE OF (managed_design_object);
    oriented_relationship : runout_tolerance_zone_orienting_relationship;
    semi_angle_value      : angle_data_element;
  INVERSE
    zone_definition : runout_tolerance_zone_definition_with_specified_angle FOR
                      runout_orientation;
END_ENTITY; -- runout_zone_orientation

(*
```

4.2.799 Scalar_terminal_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Scalar_terminal_definition is a type of Functional_unit_usage_view_terminal_definition (see 4.2.411) for which the size shall be unity.

NOTE EXPRESS specification:

```
*)
ENTITY scalar_terminal_definition
  SUBTYPE OF (functional_unit_usage_view_terminal_definition);
END_ENTITY; -- scalar_terminal_definition

--tt tc1 correction for list error in analytical_model_vector_port_assignment
(*
```

4.2.800 Scalar_terminal_definition_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Scalar_terminal_definition_link is a type of Managed_design_object (see 4.2.524).

The data associated with a `Scalar_terminal_definition_link` are the following:

- `precedent_terminal_definition`;
- `subsequent_terminal_definition`.

4.2.800.1 `precedent_terminal_definition`

The `precedent_terminal_definition` specifies a `Scalar_terminal_definition` (see 4.2.799) for the `Scalar_terminal_definition_link`.

4.2.800.2 `subsequent_terminal_definition`

The `subsequent_terminal_definition` specifies a `Scalar_terminal_definition` (see 4.2.799) for the `Scalar_terminal_definition_link`.

Constraints:

UR1: The combination of `precedent_terminal_definition` and `subsequent_terminal_definition` shall be unique within a population of `Scalar_terminal_definition_link`.

WR1: The `precedent_terminal_definition` shall not be the `subsequent_terminal_definition` within a population of `Scalar_terminal_definition_link`.

NOTE EXPRESS specification:

```

*)
ENTITY scalar_terminal_definition_link
  SUBTYPE OF (managed_design_object);
  precedent_terminal_definition : scalar_terminal_definition;
  subsequent_terminal_definition : scalar_terminal_definition;
UNIQUE
  UR1: precedent_terminal_definition, subsequent_terminal_definition;
WHERE
  WR1: precedent_terminal_definition :&#60;&#62;; subsequent_terminal_definition;
END_ENTITY; -- scalar_terminal_definition_link

( *
```

4.2.801 `Scattering_parameter_data_element`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Scattering_parameter_data_element` is a

The data associated with a `Scattering_parameter_data_element` are the following:

— `additional_data`.

4.2.801.1 `additional_data`

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY scattering_parameter_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- scattering_parameter_data_element

( *
```

4.2.802 `Seating_plane`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Seating_plane` is a type of `Non_feature_shape_element` (see 4.2.560) included specifically for the purpose of providing a description of an interface plane. One primary usage in that purpose is synchronizing the two-dimensional and three-dimensional models of a package. The seating plane simulates a mating feature when referenced as an `interface_plane`. The seating plane simulates the interconnect substrate surface in the definition view of a package, where it is referenced in the role of a `package_seating_plane` or `die_seating_plane`. When represented by a three-dimensional plane the normal shall represent the mating surface. The normal vector shall be in the Z axis and shall be at the origin of the defining coordinate system for the three-dimensional case. A `Seating_plane` scope shall be a `Bare_die`, or a `Package`.

NOTE EXPRESS specification:

```
*)
ENTITY seating_plane
  SUBTYPE OF (non_feature_shape_element);
WHERE
  WR1: SIZEOF([ 'AP210_ARM.BARE_DIE',
               'AP210_ARM.PACKAGE' ] *
            TYPEOF(SELF\non_feature_shape_element.scope)) = 1;
END_ENTITY; -- seating_plane

( *
```

4.2.803 Secondary_orientation_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Secondary_orientation_feature is both a Part_feature (see 4.2.609) and a Datum_feature (see 4.2.226). A secondary_orientation_feature provides anti-rotation capability for unambiguous orientation of the part in the datum_reference_frame.

The data associated with a Secondary_orientation_feature are the following:

— associated_body_vertical_extent.

4.2.803.1 associated_body_vertical_extent

An associated_body_vertical_extent specifies the role of the Package_body_surface (see 4.2.578) for a Secondary_orientation_feature. The associated_body_vertical_extent helps to identify the surface or surfaces which contain the feature. An associated_body_vertical_extent need not be specified for a Secondary_orientation_feature.

NOTE EXPRESS specification:

```
*)
ENTITY secondary_orientation_feature
  SUBTYPE OF (part_feature, datum_feature);
    associated_body_vertical_extent : OPTIONAL SET [1:2] OF package_body_surface;
END_ENTITY; -- secondary_orientation_feature

( *
```

4.2.804 Security_classification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Security_classification is a type of Managed_design_object (see 4.2.524). A Security_classification is the level of confidentiality that is required for the purpose of product data protection.

The data associated with a Security_classification are the following:

- classification_date;
- classification_officer;
- classification_level;
- declassification_date;
- security_classification_approval.

4.2.804.1 classification_date

The classification_date specifies either the effective Date (see 4.2.212) or the effective Date_and_time (see 4.2.213) for the Security_classification.

4.2.804.2 classification_officer

The classification_officer specifies the Person_and_organization (see 4.2.647) responsible for the Security_classification.

4.2.804.3 classification_level

The classification_level specifies the level for the Security_classification.

4.2.804.4 declassification_date

The declassification_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) upon which the Security_classification no longer applies. The declassification_date need not be provided for a particular Security_classification.

4.2.804.5 security_classification_approval

The security_classification_approval specifies the role of the Ee_approval (see 4.2.314) for the Security_classification.

NOTE EXPRESS specification:

```

*)
ENTITY security_classification
  SUBTYPE OF (managed_design_object);
    classification_officer      : person_and_organization;
    security_classification_approval : ee_approval;
    classification_date         : date_or_date_and_time;
    classification_level        : ee_name;
    declassification_date       : OPTIONAL date_or_date_and_time;
END_ENTITY; -- security_classification
( *

```

4.2.805 Separate_geometric_tolerance_requirement

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Separate_geometric_tolerance_requirement is a type of Managed_design_object (see 4.2.524). A Separate_geometric_tolerance_requirement is a specification that indicates that a set of one or more Geometric_tolerance_with_referenced_datum_system (see 4.2.422) is to be treated as a separate requirement. That is, when geometric tolerances (excluding the composite profile refinement tolerances and composite feature relating position tolerances) reference the same datum system, all the physical features to which the geometric tolerances are applied are treated as a single pattern with regard to the geometric tolerances. If this type of interrelationship is not desired, the Separate_geometric_tolerance_requirement may be used to specify a set of one or more Geometric_tolerance (see 4.2.419) that are to be treated as a separate requirement.

NOTE Separate requirements are discussed in 4.5.12 of ASME Y14.5M.

The data associated with a Separate_geometric_tolerance_requirement are the following:

— separated_geometric_tolerances.

4.2.805.1 separated_geometric_tolerances

The separated_geometric_tolerances specifies the Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that are to be treated as a separate requirement. There shall be one or more Geometric_tolerances_with_referenced_datum_system for a Separate_geometric_tolerance_requirement.

NOTE If the goal is for an individual Geometric_tolerance to be a separate requirement, then it should be specified as the separated_geometric_tolerances by a Separate_geometric_tolerance_requirement that only specifies that Geometric_tolerance.

Constraints:

WR1: The Geometric_tolerance_with_referenced_datum_system that are specified as the separated_geometric_tolerances shall all specify the same instance of Datum_system_defining_conditions as their datum_system_definition.

WR2: The Geometric_tolerance_with_referenced_datum_system that are specified as the separated_geometric_tolerances shall not either be Composite_linear_profile_refinement_tolerance, Composite_surface_profile_refinement_tolerance, or Composite_feature_relating_position_tolerance.

WR3: The Geometric_tolerance_with_referenced_datum_system that are specified as the separated_geometric_tolerances shall either be Profile_of_any_line_tolerance, Profile_of_any_surface_tolerance, or Position_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY separate_geometric_tolerance_requirement
  SUBTYPE OF (managed_design_object);
    separated_geometric_tolerances : SET [1:?] OF
                                          geometric_tolerance_with_referenced_datum_system;
END_ENTITY; -- separate_geometric_tolerance_requirement

( *
```

4.2.806 Shape_and_ee_product_definition_based_constraint_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_and_ee_product_definition_based_constraint_occurrence is a type of Ee_product_definition_based_constraint_occurrence (see 4.2.324) that has a basis in the characteristics of one or more Shape_definition (see 4.2.808) within the context of the constraining_definition inherited from Ee_product_definition_based_constraint_occurrence. Each member of Shape_definition referenced shall reference the Ee_product_definition referenced by the constraining_definition attribute inherited from Ee_product_definition_based_constraint_occurrence.

NOTE This provides a mechanism to specify that one or more of the analysis or keepout shapes on Physical_unit is a constraint.

The data associated with a Shape_and_ee_product_definition_based_constraint_occurrence are the following:

— constraining_shape.

4.2.806.1 constraining_shape

The `constraining_shape` specifies the role of the `Shape_definition` (see 4.2.808) for the `Shape_and_ee_product_definition_based_constraint_occurrence`. The interpretation shall be that the `constraining_shape` specifies the `shape_representation` and the `constraining_definition` inherited from the supertype specifies the `product_definition`. There may be one or more `Shape_definition` for the `Shape_and_ee_product_definition_based_constraint_occurrence`.

NOTE EXPRESS specification:

```

*)
ENTITY shape_and_ee_product_definition_based_constraint_occurrence
  SUBTYPE OF (ee_product_definition_based_constraint_occurrence);
  constraining_shape : SET [1:?] OF shape_definition;
WHERE
  WR1: valid_constraining_shape(constraining_shape,
    SELF\ee_product_definition_based_constraint_occurrence.
    constraining_part);
END_ENTITY; -- shape_and_ee_product_definition_based_constraint_occurrence

( *
```

4.2.807 Shape_composition_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Shape_composition_relationship` is a type of `Geometric_representation_relationship` (see 4.2.418). A `Shape_composition_relationship` is the association of a `Non_feature_shape_definition` (see 4.2.559) with its `Axis_placement` (see 4.2.56) to create a resultant `Non_feature_shape_definition` (see 4.2.559).

NOTE A `Shape_composition_relationship` is included to support composition of `Non_feature` and non-physical shapes. `Component_2d_location` and other types of location Application Objects support composition of assembly or interconnect shapes. `Usage_concept_usage_view_relationship` supports composition of shapes in the usage view. The name is generic to support extensibility. Mirroring is possible by use of `Cartesian_transformation_operator`. The use of mirroring should be carefully considered to avoid confusion in receiving systems.

EXAMPLE 1 A `Shape_composition_relationship` would be used to transform a keepout shape for a detail into an overall keepout shape. It may be that the detail shape is a `Physical_unit_planar_keepout_shape`.

The data associated with a `Shape_composition_relationship` are the following:

- `related_feature_shape`;
- `related_feature_shape_placement`;
- `resultant_feature_shape`.

4.2.807.1 related_feature_shape

The related_feature_shape specifies one role of the Non_feature_shape_definition (see 4.2.559) for the Feature_shape_composition_relationship. This role name is redeclared from the shape_2 attribute inherited from Geometric_representation_relationship.

4.2.807.2 related_feature_shape_placement

The related_feature_shape_placement specifies the role of the Axis_placement (see 4.2.56) or of the Cartesian_transformation_operator (see ??) for the Shape_composition_relationship.

4.2.807.3 resultant_feature_shape

The resultant_feature_shape specifies one role of the Non_feature_shape_definition (see 4.2.559) for the Shape_composition_relationship. This role name is redeclared from the shape_1 attribute inherited from Geometric_representation_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY shape_composition_relationship
  SUBTYPE OF (geometric_representation_relationship);
    related_feature_shape_placement : axis_placement_or_cto2d;
DERIVE
  related_feature_shape           : non_feature_shape_definition :=
    SELF\geometric_representation_relationship.shape_2;
  resultant_feature_shape         : non_feature_shape_definition :=
    SELF\geometric_representation_relationship.shape_1;
END_ENTITY; -- shape_composition_relationship
( *
```

4.2.808 Shape_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_definition is a type of Managed_design_object (see 4.2.524) A Shape_definition is one of a Bound_volume_shape (see 4.2.76), a Curve (see 4.2.203), an Edge_based_2d_wireframe (see 4.2.309), a Geometrically_bounded_2d_wireframe (see 4.2.424), an Open_shell_based_surface (see 4.2.566), a Planar_shape (see 4.2.678), or a Shell_based_2d_wireframe (see 4.2.819). A Shape_definition is also one of a Feature_shape_definition (see 4.2.372) or of a Non_feature_shape_definition (see 4.2.559). A Shape_definition is the aggregation of geometry elements that represents the shape of an object or that represents additional properties of an object or that represents annotation information. The dimensionality of the geometric elements is determined by the dimensionality of the object being represented.

The data associated with a Shape_definition are the following:

— geometric_context.

4.2.808.1 geometric_context

The geometric_context specifies the role of the Cartesian_coordinate_system (see 4.2.86) for the Shape_definition.

NOTE EXPRESS specification:

```

*)
ENTITY shape_definition
  ABSTRACT SUPERTYPE OF ((ONEOF (
    bound_volume_shape,
    open_shell_based_surface,
    manifold_subsurface_shape,
    planar_shape)) AND
    (ONEOF(feature_shape_definition,
      non_feature_shape_definition)))
  SUBTYPE OF (managed_design_object);
  geometric_context : cartesian_coordinate_system;
  element : OPTIONAL SET [1:?] OF shape_item;
END_ENTITY; -- shape_definition
( *
```

4.2.809 Shape_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_dimension_representation is a type of Managed_design_object (see 4.2.524). A Shape_dimension_representation is one of an Externally_defined_dimension_representation (see 4.2.363), a Limit_dimension_representation (see 4.2.514), a Plus_minus_tolerance_range_representation (see 4.2.692), or a Singular_dimension_representation (see 4.2.829). A Shape_dimension_representation is a representation of a Dimensional_characteristic (see 4.2.298). The Shape_dimension_representation explicitly describes one or more Dimensional_characteristic with a single value, a range of values, a combination of values, or values and references to the standard documents that define the meaning of these values.

Constraints:

WR1: The Shape_dimension_representation shall be specified as a representation of at least one Dimensional_characteristic_representation.

NOTE EXPRESS specification:

```

*)
ENTITY shape_dimension_representation
  ABSTRACT SUPERTYPE OF (ONEOF (limit_dimension_representation,
    singular_dimension_representation,
    externally_defined_dimension_representation,
    plus_minus_tolerance_range_representation))
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- shape_dimension_representation

( *

```

4.2.810 Shape_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element is a type of Managed_design_object (see 4.2.524). A Shape_element is one of a Non_feature_shape_element (see 4.2.560) or a Physical_feature_or_part_template (see 4.2.658) or a Land_template_terminal (see 4.2.496) or a Component_termination_passage_template_terminal (see 4.2.153) or a Via_template_terminal (see 4.2.953). A Shape_element may additionally be a Composite_shape_element (see 4.2.162). A Shape_element is an identifiable element of the shape of a Physical_unit (see 4.2.664).

The data associated with a Shape_element are the following:

- description;
- name.

4.2.810.1 description

The description specifies the label that conveys the nature of the Shape_element. The description need not be specified for a Shape_element.

4.2.810.2 name

The name specifies the identifier of the Shape_element.

NOTE EXPRESS specification:

```

*)
ENTITY shape_element
  ABSTRACT SUPERTYPE OF (ONEOF (physical_feature_or_part_template,
    component_termination_passage_template_terminal,
    land_template_terminal,
    via_template_terminal,
    non_feature_shape_element) ANDOR composite_shape_element)
  SUBTYPE OF (managed_design_object);
  name          : label;
  description : OPTIONAL label;
WHERE
  WR1: NOT('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) OR
    (SIZEOF(TYPEOF(SELF)) &#62; 2);
  WR2: NOT(('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND
    ('AP210_ARM.COMPONENT_TERMINATION_PASSAGE_TEMPLATE_TERMINAL' IN
    TYPEOF(SELF)));
  WR3: NOT(('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND
    ('AP210_ARM.LAND_TEMPLATE_TERMINAL' IN
    TYPEOF(SELF)));
  WR4: NOT(('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND
    ('AP210_ARM.VIA_TEMPLATE_TERMINAL' IN
    TYPEOF(SELF)));
END_ENTITY; -- shape_element

( *

```

4.2.811 Shape_element_boundary_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_boundary_relationship is a type of Shape_element_relationship (see 4.2.816) that specifies that a Shape_element (see 4.2.810) is the boundary or a portion of the boundary of another Shape_element. The related_shape_element specifies the Shape_element that is the boundary or a portion of a boundary of the Shape_element specified as the relating_shape_element.

NOTE 1 The attribute related_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

NOTE 2 The attribute relating_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

EXAMPLE 1 A Shape_element_boundary_relationship may be used to specify that a Shape_element is one of the boundaries of a rectangular Datum_target (see 4.2.240) area. The boundary may then be located relative to a Datum (see 4.2.214) with a Dimensional_location (see 4.2.300).

Constraints:

WR1: The Shape_element specified as the relating_shape_element shall not be specified as a boundary of itself either directly or through intermediate Shape_element_boundary_relationship.

NOTE EXPRESS specification:

```
*)
ENTITY shape_element_boundary_relationship
  SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- shape_element_boundary_relationship
```

(*

4.2.812 Shape_element_composing_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_composing_relationship is a type of Shape_element_constituent_relationship (see 4.2.813) that associates a Composite_shape_element (see 4.2.162) with a Shape_element (see 4.2.810) from which the Composite_shape_element is composed.

The data associated with a Shape_element_composing_relationship are the following:

— relating_shape_element.

4.2.812.1 relating_shape_element

The relating_shape_element specifies the Composite_shape_element (see 4.2.162) that is the Shape_element (see 4.2.810) that is composed from the Shape_element specified as the related_shape_element for the Shape_element_composing_relationship.

NOTE The attribute related_shape_element is declared in the entity Shape_element_relationship of which this entity is a subtype.

NOTE EXPRESS specification:

```
*)
ENTITY shape_element_composing_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relatating_shape_element : composite_shape_element;
END_ENTITY; -- shape_element_composing_relationship
```

(*

4.2.813 Shape_element_constituent_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_constituent_relationship is a type of Shape_element_relationship (see 4.2.816) that associates a Shape_element (see 4.2.810) with a second Shape_element to which the first Shape_element is a constituent. The Shape_element that is the constituent Shape_element is specified as the related_shape_element. The Shape_element to which the constituent Shape_element is a constituent is specified as the relating_shape_element. A Shape_element_constituent_relationship may be a Shape_element_composing_relationship (see 4.2.812).

NOTE 1 The attribute relating_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

NOTE 2 The attribute related_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

NOTE 3 The difference between a Shape_element_constituent_relationship and a Shape_element_composing_relationship is that the Shape_element_constituent_relationship is used to indicate that a Shape_element is part of another Shape_element. However, the Shape_element_composing_relationship is used to completely specify the composition of a Composite_shape_element.

EXAMPLE 1 A Shape_element_constituent_relationship might be used to indicate that a Datum_target is on the bottom surface of a printed circuit board. In this case, the Shape_element that is the Datum_target would be specified as the related_shape_element, and the Shape_element that is the bottom surface of the printed circuit board would be specified as the relating_shape_element.

Constraints:

UR1: The combination of the relating_shape_element inherited from Shape_element_relationship and related_shape_element inherited from Shape_element_relationship shall be unique within a population of Shape_element_constituent_relationship. That is, any specific Shape_element shall not be a constituent of any one Shape_element more than once.

WR1: The Shape_element that is specified as the relating_shape_element shall not be a constituent of itself, either directly or indirectly.

NOTE EXPRESS specification:

```

*)
ENTITY shape_element_constituent_relationship
  SUPERTYPE OF (shape_element_composing_relationship)
  SUBTYPE OF (shape_element_relationship);
  UNIQUE
    UR1 : related_shape_element, relating_shape_element;
END_ENTITY; -- shape_element_constituent_relationship

( *

```

4.2.814 Shape_element_deriving_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_deriving_relationship is a type of Shape_element_relationship (see 4.2.816) that associates a Derived_shape_element (see 4.2.260) with a Shape_element (see 4.2.810) from which the Derived_shape_element is derived.

The data associated with a Shape_element_deriving_relationship are the following:

— relating_shape_element.

4.2.814.1 relating_shape_element

The relating_shape_element specifies the Derived_shape_element (see 4.2.260) that is the Shape_element (see 4.2.810) that is derived from the Shape_element specified as the related_shape_element for the Shape_element_deriving_relationship.

NOTE The attribute related_shape_element is declared in the Shape_element_relationship entity of which this entity is a subtype.

EXAMPLE 1 Consider a centre axis that is derived from a cylindrical hole. The nature of the relationship between the centre axis and the cylindrical hole is indicated by a Shape_element_deriving_relationship in which the cylindrical hole is specified as the related_shape_element and the centre axis is specified as the relating_shape_element.

Constraints:

UR1: The combination of the relating_shape_element inherited from Shape_element_relationship and related_shape_element inherited from Shape_element_relationship shall be unique within a population of Shape_element_deriving_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY shape_element_deriving_relationship
  SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.relate_shape_element : derived_shape_element;
  UNIQUE
  UR1 : related_shape_element, relating_shape_element;
END_ENTITY; -- shape_element_deriving_relationship

( *

```

4.2.815 Shape_element_locating_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_locating_relationship is a type of Managed_design_object (see 4.2.524). A Shape_element_locating_relationship is the association between two Shape_element (see 4.2.810) in which the associated Shape_elements are located in relation to each other.

The data associated with a Shape_element_locating_relationship are the following:

- located_shape_element;
- locating_shape_element.

4.2.815.1 located_shape_element

The located_shape_element specifies the Shape_element (see 4.2.810) that is located in relation to the Shape_element specified by the locating_shape_element for the Shape_element_locating_relationship.

4.2.815.2 locating_shape_element

The locating_shape_element specifies the Shape_element (see 4.2.810) in relation to which the Shape_element specified by the located_shape_element is located for the Shape_element_locating_relationship.

Constraints:

UR1: The combination of located and locating Shape_elements shall be unique within a population of Shape_element_locating_relationship. That is, the same Shape_elements shall not be located relative to each other more than once.

WR1: The Shape_element specified by the located_shape_element shall not be the same Shape_element specified by the locating_shape_element.

WR2: The Shape_element_locating_relationship shall be specified as either the located_elements by a Dimensional_location or as the boundary_separation by a Dimensional_size_based_on_opposing_boundaries.

NOTE EXPRESS specification:

```

*)
ENTITY shape_element_locating_relationship
  SUBTYPE OF (managed_design_object);
    locating_shape_element : shape_element;
    located_shape_element  : shape_element;
  UNIQUE
    UR1 : located_shape_element, locating_shape_element;
END_ENTITY; -- shape_element_locating_relationship

( *
```

4.2.816 Shape_element_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shape_element_relationship is a type of Managed_design_object (see 4.2.524). A Shape_element_relationship is one of a Parallel_shape_element_relationship (see 4.2.601), a Perpendicular_shape_element_relationship (see 4.2.644), a Shape_element_boundary_relationship (see 4.2.811), a Shape_element_constituent_relationship (see 4.2.813), a Non_feature_shape_element_relationship (see 4.2.561), or a Shape_element_deriving_relationship (see 4.2.814). Additionally, a Shape_element_relationship may be a Datum_feature_usage_in_datum_relationship (see 4.2.229). A Shape_element_relationship is an association between two Shape_element (see 4.2.810).

The data associated with a Shape_element_relationship are the following:

- description;
- name;
- related_shape_element;
- relating_shape_element.

4.2.816.1 description

The description specifies the descriptive label that relates the nature of the Shape_element_relationship. The description need not be specified for a Shape_element_relationship.

4.2.816.2 name

The name specifies the identifying label for the Shape_element_relationship. The name need not be specified for a Shape_element_relationship.

4.2.816.3 related_shape_element

The related_shape_element specifies the Shape_element (see 4.2.810) to which the relating_shape_element refers for the Shape_element_relationship. If one Shape_element of the relationship is dependent upon the other, this attribute shall be the dependent one.

4.2.816.4 relating_shape_element

The relating_shape_element specifies the Shape_element (see 4.2.810) that relates to another, for the Shape_element_relationship.

Constraints:

WR1: If the Shape_element_relationship is a Datum_feature_usage_in_datum_relationship it shall not also be a Parallel_shape_element_relationship, a Perpendicular_shape_element_relationship, a Shape_element_boundary_relationship, or a Shape_element_constituent_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY shape_element_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (parallel_shape_element_relationship,
    perpendicular_shape_element_relationship,
    datum_difference,
    non_feature_shape_element_relationship,
    shape_element_boundary_relationship,
    shape_element_constituent_relationship,
    shape_element_deriving_relationship) ANDOR
    datum_feature_usage_in_datum_relationship)
  SUBTYPE OF (managed_design_object);
    relating_shape_element : shape_element;
    description             : OPTIONAL label;
    related_shape_element  : shape_element;
    name                    : OPTIONAL label;
END_ENTITY; -- shape_element_relationship

```

(*

4.2.817 Shape_formed_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Shape_formed_terminal` is a type of a `Altered_package_terminal` (see 4.2.9) that defines a modification of its shape from its original `Package_terminal` (see 4.2.581) shape. This shape modification is done in order to facilitate the installation of a `Packaged_part` (see 4.2.591) in an assembly.

The data associated with a `Shape_formed_terminal` are the following:

— `terminal_form_specification`.

4.2.817.1 terminal_form_specification

The `terminal_form_specification` specifies the `Lead_form_specification` (see 4.2.510) of the `Shape_formed_terminal`.

NOTE EXPRESS specification:

```
*)
ENTITY shape_formed_terminal
  SUBTYPE OF (altered_package_terminal);
    terminal_form_specification : lead_form_specification;
END_ENTITY; -- shape_formed_terminal

--tt tc1 wd9 modified
(*
```

4.2.818 Shape_item

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Shape_item` is a type of `Managed_design_object` (see 4.2.524). A `Shape_item` may be one of a `Boolean_csg_result` (see 4.2.75), a `Cartesian_point` (see 4.2.87), a `Curve` (see 4.2.203), a `Primitive_2d` (see 4.2.720), a `Axis_placement` (see 4.2.56), or additionally may be a `Styled_text` (see 4.2.876). The `Shape_item` shall not be a complex instance of `Styled_text` and `Cartesian_point`. The `Shape_item` shall not be a complex instance of `Styled_text` and `Axis_placement`.

NOTE EXPRESS specification:

```

*)
ENTITY shape_item
  SUPERTYPE OF ((ONEOF (
    boolean_csg_result,
    cartesian_point,
    curve,
    axis_placement,
    primitive_2d)) ANDOR styled_text)
  SUBTYPE OF (managed_design_object);
DERIVE
  geometric_context : cartesian_coordinate_system :=
    associated_shape_definition[1].geometric_context;
INVERSE
  associated_shape_definition : SET [1:?] OF shape_definition
    for element;
WHERE
  WR1: NOT (('AP210_ARM.CARTESIAN_POINT' IN TYPEOF(SELF))
    AND
    ('AP210_ARM.STYLED_TEXT' IN TYPEOF(SELF)));
  WR2: NOT (('AP210_ARM.AXIS_PLACEMENT' IN TYPEOF(SELF))
    AND
    ('AP210_ARM.STYLED_TEXT' IN TYPEOF(SELF)));
END_ENTITY; -- shape_item
( *

```

4.2.819 Shell_based_2d_wireframe

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shell_based_2d_wireframe is a type of Feature_shape_definition (see 4.2.372). A Shell_based_2d_wireframe describes the shape of a product by two-dimensional wireframe geometry with topology. This includes all two-dimensional curves and topological entities that define a graph of vertices, edges, and loops.

NOTE EXPRESS specification:

```

*)
ENTITY shell_based_2d_wireframe
  SUBTYPE OF (planar_shape);
END_ENTITY; -- shell_based_2d_wireframe
( *

```

4.2.820 Shell_based_wireframe

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Shell_based_wireframe` is a type of `Bound_volume_shape` (see 4.2.76). A `Shell_based_wireframe` maintains the topology information along with the geometric wireframe model.

NOTE EXPRESS specification:

```
*)
ENTITY shell_based_wireframe
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- shell_based_wireframe

( *
```

4.2.821 Shield

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Shield` is a type of `Assembly_component` (see 4.2.34) included for suppression of unwanted electromagnetic coupling among components. An `Assembly_component` included in the assembly for other purposes might also serve as a `Shield`, and in that case the other type would be specified as well for that instance.

EXAMPLE 1 A `Via` (see 4.2.947) may be a `Shield`. More than one shield may shield the same `shielded_item` for the same purpose.

EXAMPLE 2 Figure 101 illustrates a detail of a layout drawing that shows a shielded trace (carrying a critical signal) on the several layers it traverses in the design.

The data associated with a `Shield` are the following:

— `shielded_item`.

4.2.821.1 shielded_item

The `shielded_item` specifies either an `Assembly_component` (see 4.2.34), a `Component_feature` (see 4.2.118), or a `Physical_connectivity_definition` (see 4.2.652) as the shieldable item for a `Shield`. There shall be one or more `Assembly_component`, `Component_feature`, or `Physical_connectivity_definition` for a `Shield`.

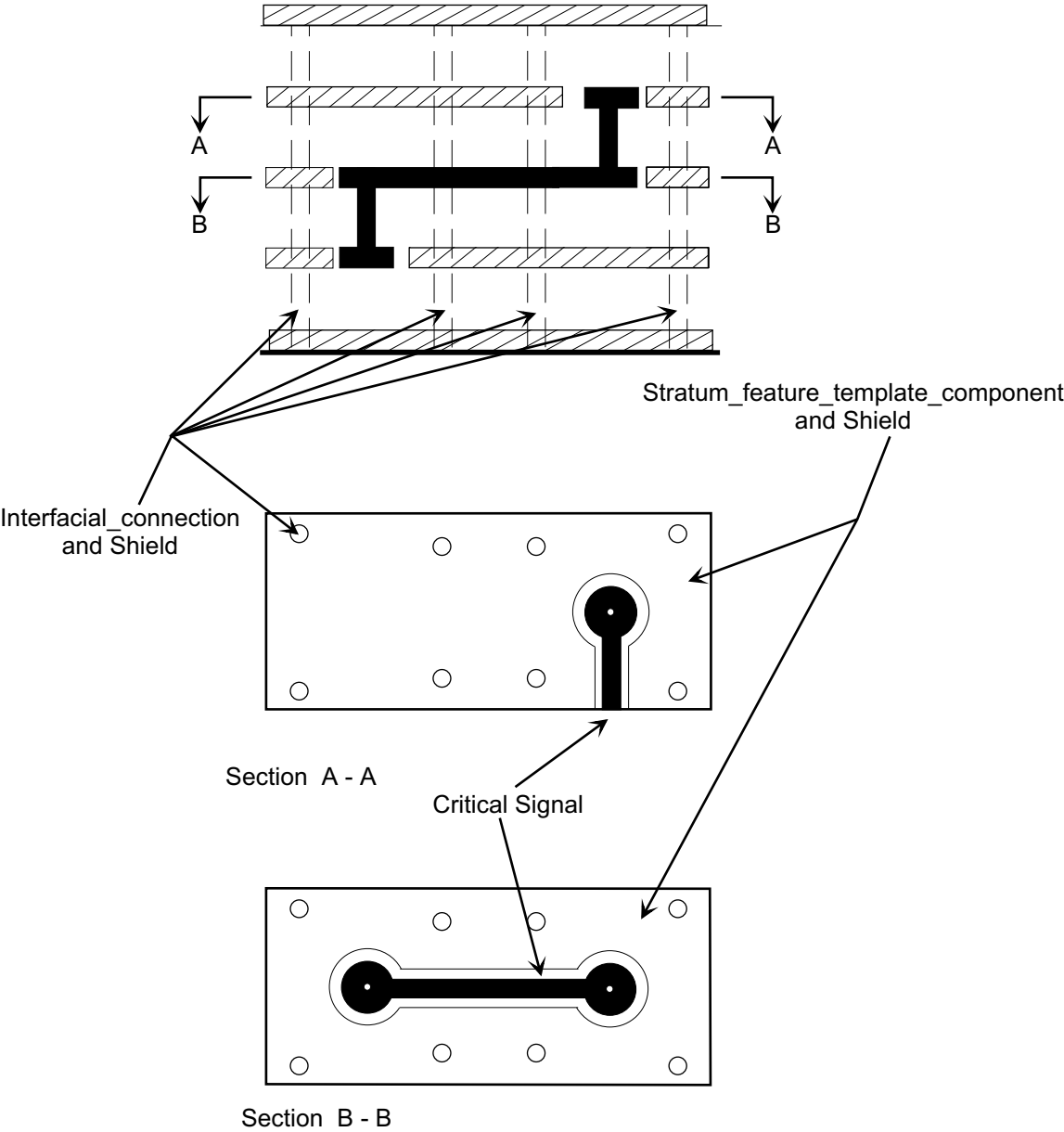


Figure 101 – Shield

NOTE EXPRESS specification:

```
*)  
ENTITY shield  
  SUBTYPE OF (assembly_component);  
    shielded_item : SET [1:?] OF shieldable_item;  
END_ENTITY; -- shield  
  
(*
```

4.2.822 Shield_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Shield_allocation is a type of Electromagnetic_compatibility_requirement_allocation (see 4.2.347). The Shield_allocation enables the assignment of a requirement to a shield. This allocates required properties of shielding effectiveness to the shield.

The data associated with a Shield_allocation are the following:

- assembly_assignment;
- requirement.

4.2.822.1 assembly_assignment

The assembly_assignment specifies the role of the Shield (see 4.2.821) for the Shield_allocation.

4.2.822.2 requirement

The requirement specifies the role of the Electromagnetic_requirement_occurrence (see 4.2.348) for the Shield_allocation.

NOTE EXPRESS specification:

```
*)  
ENTITY shield_allocation  
  SUBTYPE OF (electromagnetic_compatibility_requirement_allocation);  
    SELF\assembly_requirement_allocation.assembly_assignment : shield;  
    SELF\assembly_requirement_allocation.requirement          :  
electromagnetic_requirement_occurrence;  
END_ENTITY; -- shield_allocation  
  
(*
```

4.2.823 Signal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Signal is a type of Managed_design_object (see 4.2.524). A Signal in this part of ISO 10303 is an abstraction of signal as categorized according to an ISO or IEC standard. Signal in this part of ISO 10303 is specifically included for the purpose of exchanging information related to functional abstractions of products as commonly specified on data sheets. There is no predefined mechanism in this part of ISO 10303 that defines standard physical arrangement of instrumentation and test articles to support test processes. This part of ISO 10303 provides a standard mechanism to associate sets of data that characterize the behaviour of a product with the functional hierarchical model and physical geometric model of that product.

NOTE 1 A Signal is a physical phenomenon that may be characterized by means of appropriate instrumentation or simulated by a computer system for the purpose of evaluating system behaviour.

NOTE 2 IEC Multi-Lingual Dictionary Volume 2, 1992: signal (174) a physical phenomenon, one or more of whose characteristics may vary to represent information.

EXAMPLE 1 The physical phenomenon may be, for instance, an electromagnetic wave, or acoustic wave and the characteristic may be an electric field, a voltage, or a sound pressure.

The data associated with a Signal are the following:

— signal_category.

4.2.823.1 signal_category

The signal_category specifies the External_definition (see 4.2.361) for the category for this Signal.

EXAMPLE 1 Examples of signal_category include an electromagnetic wave, or acoustic wave.

NOTE EXPRESS specification:

```
*)
ENTITY signal
  SUBTYPE OF (managed_design_object);
  signal_category : external_definition;
END_ENTITY; -- signal

( *
```

4.2.824 Signal_property

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Signal_property` is a type of `Managed_design_object` (see 4.2.524). A `Signal_property` may be a `Composite_signal_property` (see 4.2.163). A `Signal_property` is a data set that characterizes a `Signal` (see 4.2.823) and each element of which has units that are traceable to an ISO or IEC standard. The units may be basic SI units, derived SI units, conversion based units, or context dependent units. Conversion based and context dependent units shall be specified in the external standard that defines the `signal_property_category`.

The data associated with a `Signal_property` are the following:

- `characterized_signal`;
- `signal_property_category`;
- `signal_property_representation`.

4.2.824.1 characterized_signal

The `characterized_signal` specifies the `Signal` (see 4.2.823) that has the `Signal_property`.

4.2.824.2 signal_property_category

The `signal_property_category` specifies the role of the `External_definition` (see 4.2.361) for the `Signal_property`.

4.2.824.3 signal_property_representation

The `signal_property_representation` specifies the `Characteristic_data_table` (see 4.2.100) of the `Signal_property`. The `signal_property_representation` is at least a two dimensional representation, including a sample function and a data set represented at the discrete points of the sample function.

NOTE A `signal_property_representation` is a discretized function represented by a table that maps a parameter, such as time, frequency, or distance, onto a response variable. The table is represented in this standard by a list of columnar data, each column is represented by set. One member of the set is a representation of the sample function at that sample point. The other members of the set represents the discretized parameters (e.g., power).

EXAMPLE 1 An example `signal_property_representation` characterizes voltage over a defined frequency range.

NOTE EXPRESS specification:

```

*)
ENTITY signal_property
  SUPERTYPE OF (composite_signal_property)
  SUBTYPE OF (managed_design_object);
  signal_property_representation : characteristic_data_table;
  characterized_signal           : signal;
  signal_property_category       : external_definition;
END_ENTITY; -- signal_property

( *

```

4.2.825 Signal_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Signal_specification is a type of Managed_design_object (see 4.2.524). A Signal_specification is a bounded behavioural characteristic as measured between two terminals of a Functional_unit (see 4.2.400) in a controlled environment. The characterization of the behaviour of the functional_unit_definition is in terms of the representations of the Signals that exist at the interface of the instance of the functional_unit_definition with the network within which the instance is embedded.

The data associated with a Signal_specification are the following:

- characterization_network;
- characterizing_functional_unit;
- maximum_characteristic;
- measurement_access_mechanism;
- measurement_condition;
- minimum_characteristic;
- reference_access_mechanism.

4.2.825.1 characterization_network

The characterization_network specifies the Functional_unit_network_definition (see 4.2.402) in which an instance of a usage view of the Functional_unit_definition (see 4.2.401) being characterized is embedded for this Signal_specification. In this part of ISO 10303, the characterization_network shall include all required simulated stimulus sources and measurement sensors.

EXAMPLE 1 Examples of characterization_network types include test bench (VHDL), test fixture, test circuit.

4.2.825.2 characterizing_functional_unit

The `characterizing_functional_unit` specifies the `Functional_unit` (see 4.2.400) being characterized by the `Signal_specification`.

4.2.825.3 maximum_characteristic

The `maximum_characteristic` specifies the upper bound for the valid range of the `Signal_property` (see 4.2.824) for the `Signal_specification`.

4.2.825.4 measurement_access_mechanism

The `measurement_access_mechanism` specifies the role of the `Functional_unit_terminal` (see 4.2.407) for the `Signal_specification`.

4.2.825.5 measurement_condition

The `measurement_condition` specifies the `Test_specification` (see 4.2.898) under which the measurements are valid, including allowed ranges for stimulus conditions, for the `Signal_specification`.

NOTE The `measurement_condition` may specify data using text or graphics.

4.2.825.6 minimum_characteristic

The `minimum_characteristic` specifies the lower bound for the valid range of the `Signal_property` (see 4.2.824) for the `Signal_specification`.

4.2.825.7 reference_access_mechanism

The `reference_access_mechanism` specifies the `Functional_unit_terminal` (see 4.2.407) on the `Functional_unit` (see 4.2.400) being characterized by this `Signal_specification`. The `reference_access_mechanism` need not be specified for a particular `Signal_specification`. The purpose of the `reference_access_mechanism` is to provide a reference facility for measurements. The `reference_access_mechanism` need not be provided.

EXAMPLE 1 The `reference_access_mechanism` need not be supplied for evaluation of non-causal systems such as s- or z-plane characterized control system elements or digital elements. There is no assumption in this part of ISO 10303 that `reference_access_mechanism` is related to any standard reference node, for example, "0" in certain continuous system simulators. Example names of the terminal specified by `reference_access_mechanism` include Ground, Analogue ground, Digital ground, Trigger in, Gate out.

Constraints:

WR1: The External_definition in the role of signal_property_category specified by signal_property in the role of maximum_characteristic shall be the same as the External_definition in the role of signal_property_category specified by signal_property in the role of minimum_characteristic.

WR2: The measurement_access_mechanism shall access an occurrence of the definition being characterized.

WR3: The measurement_access_mechanism shall not be the reference_access_mechanism.

WR4: The network in which the functional_unit_definition being characterized is instantiated shall not be the network being characterized.

WR5: The characterizing_functional_unit shall help compose the characterization_network.

WR6: The measurement_access_mechanism shall be an access_mechanism for the characterizing_functional_unit.

WR7: The reference_access_mechanism, if provided, shall be an access_mechanism for the characterizing_functional_unit.

NOTE EXPRESS specification:

*)

ENTITY signal_specification

SUBTYPE OF (managed_design_object);

measurement_condition	: test_specification;
measurement_access_mechanism	: functional_unit_terminal;
reference_access_mechanism	: OPTIONAL functional_unit_terminal;
minimum_characteristic	: signal_property;
maximum_characteristic	: signal_property;
characterization_network	: functional_unit_network_definition;
characterizing_functional_unit	: functional_unit;

END_ENTITY; -- signal_specification

(*

4.2.826 Simultaneous_geometric_tolerance_requirement

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Simultaneous_geometric_tolerance_requirement is a type of Managed_design_object (see 4.2.524). A Simultaneous_geometric_tolerance_requirement is a specification that a set of two or more Geometric_tolerance (see 4.2.419) is to be treated as a simultaneous requirement. That is, the shape elements to which composite profile refinement tolerances and composite feature relating position tolerances are applied are treated as separate shape elements with regard to the geometric tolerances. If this type of independence is not desired, the Simultaneous_geometric_tolerance_requirement may be used to specify a set of two or more Geometric_tolerance that are to be treated as a simultaneous requirement.

NOTE Simultaneous requirements are discussed in 4.5.12.1 of ASME Y14.5M.

The data associated with a Simultaneous_geometric_tolerance_requirement are the following:

— grouped_geometric_tolerances.

4.2.826.1 grouped_geometric_tolerances

The grouped_geometric_tolerances specifies the Geometric_tolerance (see 4.2.419) that are to be treated as a simultaneous requirement. There shall be two or more Geometric_tolerance for a Simultaneous_geometric_tolerance_requirement.

Constraints:

WR1: The Geometric_tolerance that are specified as the grouped_geometric_tolerances either shall all be Geometric_tolerance_without_referenced_datum_system or shall all specify the same instance of Datum_system_defining_conditions as their datum_system_definition.

WR2: The Geometric_tolerance that are specified as the grouped_geometric_tolerances shall either be Composite_linear_profile_refinement_tolerance, Composite_surface_profile_refinement_tolerance, or Composite_feature_relating_position_tolerance.

NOTE EXPRESS specification:

```
*)
ENTITY simultaneous_geometric_tolerance_requirement
  SUBTYPE OF (managed_design_object);
    grouped_geometric_tolerances : SET [2:?] OF geometric_tolerance;
END_ENTITY; -- simultaneous_geometric_tolerance_requirement

( *
```

4.2.827 Single_boundary_dimension

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Single_boundary_dimension is a type of Dimensional_size (see 4.2.301) that characterizes the size of a Shape_element (see 4.2.810) that can be considered to have a single boundary or have the form of a curve. A Single_boundary_dimension may be a Curve_dimension (see 4.2.204).

EXAMPLE 1 The diameter of a cylindrical Shape_element would be characterized with a Single_boundary_dimension.

NOTE EXPRESS specification:

```

*)
ENTITY single_boundary_dimension
  SUPERTYPE OF (curve_dimension)
  SUBTYPE OF (dimensional_size);
END_ENTITY; -- single_boundary_dimension

( *

```

4.2.828 Single_datum

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Single_datum is a Datum (see 4.2.214) that is established by a single Datum_feature (see 4.2.226).

NOTE The establishment of a datum by a single datum feature is described in 6.2.1 of ISO 5459.

Constraints:

WR1: Each Single_datum shall participate in establishing at least one Datum_system.

WR2: Each Single_datum shall be specified as the relating_shape_element by exactly one Datum_feature_usage_in_single_datum_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY single_datum
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (datum);
END_ENTITY; -- single_datum

( *

```

4.2.829 Singular_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Singular_dimension_representation is a type of Shape_dimension_representation (see 4.2.809) that explicitly describes one or more Dimensional_characteristic (see 4.2.298) with a single value. A Singular_dimension_representation is one of a Maximum_dimension_representation (see 4.2.541), a Minimum_dimension_representation (see 4.2.547), or a Theoretically_exact_dimension_representation (see 4.2.899).

The data associated with a Singular_dimension_representation are the following:

— dimension_value.

4.2.829.1 dimension_value

The dimension_value specifies the Data_element (see 4.2.211) that describes the Dimensional_characteristic (see 4.2.298) for which the Singular_dimension_representation is a representation.

NOTE EXPRESS specification:

```
*)  
ENTITY singular_dimension_representation  
  ABSTRACT SUPERTYPE OF (ONEOF (  
    theoretically_exact_dimension_representation,  
    maximum_dimension_representation,  
    minimum_dimension_representation))  
  SUBTYPE OF (shape_dimension_representation);  
    dimension_value : data_element;  
END_ENTITY; -- singular_dimension_representation  
  
(*
```

4.2.830 Solid_angle_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Solid_angle_data_element is a

The data associated with a Solid_angle_data_element are the following:

— additional_data.

4.2.830.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY solid_angle_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- solid_angle_data_element

( *
```

4.2.831 Solid_curve

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Solid_curve is a type of Managed_design_object (see 4.2.524). A Solid_curve is a Curve (see 4.2.203) that is arc-wise connected (continuous).

NOTE EXPRESS specification:

```
*)
ENTITY solid_curve
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- solid_curve

( *
```

4.2.832 Solid_fill_style

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Solid_fill_style is a type of Managed_design_object (see 4.2.524). A Solid_fill_style is a style of fill pattern that completely fills the Fill_area (see 4.2.376).

NOTE EXPRESS specification:

```
*)
ENTITY solid_fill_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- solid_fill_style

( *
```

4.2.833 Solid_of_linear_extrusion

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Solid_of_linear_extrusion` is a type of `Bound_volume_shape` (see 4.2.76) defined by sweeping a planar face. The direction of translation is defined by a direction vector normal to the plane of the swept face and the length of the translation is defined by a distance "depth". The planar face may have holes which will sweep into holes in the solid.

NOTE EXPRESS specification:

```
*)
ENTITY solid_of_linear_extrusion
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- solid_of_linear_extrusion

( *
```

4.2.834 `Special_symbol_laminate_component`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Special_symbol_laminate_component` is a type of `Laminate_component` (see 4.2.483) such as an enterprise logo, trademark, or other enterprise specific symbol. A `Special_symbol_laminate_component` may be a `Primary_stratum_indicator_symbol` (see 4.2.719).

NOTE 1 A `Special_symbol_laminate_component` may or may not be limited to a member of a library of accepted symbols.

NOTE 2 A `Special_symbol_laminate_component` may or may not be scalable. A `Special_symbol_laminate_component` may or may not be associated with a physical net.

NOTE 3 These symbols may be stored in a library for legal reasons.

The data associated with a `Special_symbol_laminate_component` are the following:

- `stratum_feature_implementation`;
- `usage_definition`.

4.2.834.1 `stratum_feature_implementation`

The `stratum_feature_implementation` specifies the role of the `Stratum_feature` (see 4.2.851) for the `Special_symbol_laminate_component`.

4.2.834.2 `usage_definition`

The `usage_definition` specifies the role of the `Special_symbol_part_template` (see 4.2.835) for the `Special_symbol_laminate_component`.

NOTE EXPRESS specification:

```

*)
ENTITY special_symbol_laminate_component
  SUPERTYPE OF (primary_stratum_indicator_symbol)
  SUBTYPE OF (laminate_component);
  SELF\assembly_component.usage_definition : special_symbol_part_template;
  stratum_feature_implementation           : stratum_feature;
END_ENTITY; -- special_symbol_laminate_component

( *

```

4.2.835 Special_symbol_part_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Special_symbol_part_template is a type of Part_template (see 4.2.620) that defines a Special_symbol.

EXAMPLE 1 A corporate logo would be a Special_symbol_part_template.

The data associated with a Special_symbol_part_template are the following:

— of_stratum_technology.

4.2.835.1 of_stratum_technology

The of_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Material_removal_feature_template.

NOTE EXPRESS specification:

```

*)
ENTITY special_symbol_part_template
  SUBTYPE OF (part_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- special_symbol_part_template

( *

```

4.2.836 Specification_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Specification_allocation` is a type of `Managed_design_object` (see 4.2.524). A `Specification_allocation` is the association between an `Ee_specification` (see 4.2.338) and one or more of an `Ee_product` (see 4.2.319), an `Ee_product_configuration` (see 4.2.322), an `Ee_product_definition` (see 4.2.323), an `Ee_product_version` (see 4.2.328), a `Physical_connectivity_definition` (see 4.2.652), a `Physical_connectivity_element` (see 4.2.654), one or more of an `Assembly_component` (see 4.2.34), a `Component_feature` (see 4.2.118), a `Stratum` (see 4.2.845), one or more of an `Inter_stratum_feature` (see 4.2.432), a `Laminate_component` (see 4.2.483), or a `Stratum_feature` (see 4.2.851) that results from the design activity.

The data associated with a `Specification_allocation` are the following:

- `allocated_specification`;
- `target_product_object`.

4.2.836.1 `allocated_specification`

The `allocated_specification` specifies the role of the `Ee_specification` (see 4.2.338) for the `Specification_allocation`.

4.2.836.2 `target_product_object`

The `target_product_object` specifies either an `Ee_product` (see 4.2.319), an `Ee_product_configuration` (see 4.2.322), an `Ee_product_definition` (see 4.2.323), an `Ee_product_version` (see 4.2.328), a `Part_feature` (see 4.2.609), a `Physical_connectivity_definition` (see 4.2.652), a `Physical_connectivity_element` (see 4.2.654), or one of an `Assembly_component` (see 4.2.34), a `Component_external_reference` (see 4.2.117), a `Component_feature` (see 4.2.118), a `Component_feature_external_reference` (see 4.2.119), a `Stratum` (see 4.2.845) or one of an `Inter_stratum_feature` (see 4.2.432), a `Laminate_component` (see 4.2.483), a `Stratum_feature` (see 4.2.851), or a `Stratum_surface` (see 4.2.868) as the target through which the `Ee_specification` (see 4.2.338) is realized by the `Specification_allocation`. There shall be one or more `Ee_product`, `Ee_product_configuration`, `Ee_product_definition`, `Ee_product_version`, `Part_feature`, `Physical_connectivity_definition`, `Physical_connectivity_element`, `Assembly_component`, `Component_external_reference`, `Component_feature`, `Component_feature_external_reference`, `Stratum`, `Inter_stratum_feature`, `Laminate_component`, `Stratum_feature`, or `Stratum_surface` for a `Specification_allocation`.

NOTE EXPRESS specification:

```

*)
ENTITY specification_allocation
  SUBTYPE OF (managed_design_object);
    allocated_specification : ee_specification;
    target_product_object   : SET [1:?] OF specification_product_object_select;
END_ENTITY; -- specification_allocation

( *
```

4.2.837 Standard_uncertainty

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Standard_uncertainty is a type of Expanded_uncertainty (see 4.2.359). A Standard_uncertainty may be an Expanded_uncertainty. The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY standard_uncertainty
  SUPERTYPE OF (expanded_uncertainty)
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY;
( *
```

4.2.838 Start_order

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Start_order is a type of Work_order (see 4.2.964) that authorizes work in the ongoing development of a product design of a product, resulting in the creation of a new Ee_product_version (see 4.2.328).

NOTE EXPRESS specification:

```
*)
ENTITY start_order
  SUBTYPE OF (work_order);
END_ENTITY; -- start_order

( *
```

4.2.839 Start_request

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Start_request is a type of Work_request (see 4.2.965) that initiates the design of a part.

NOTE EXPRESS specification:

```

*)
ENTITY start_request
  SUBTYPE OF (work_request);
END_ENTITY; -- start_request

( *
```

4.2.840 Statistical_dimensional_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Statistical_dimensional_tolerance is a type of Dimensional_characteristic (see 4.2.298) that specifies design requirements that are based on sound statistics. Associated with the Statistical_dimensional_tolerance is the requirement that a specified level of process control be maintained during the manufacture of the Physical_unit (see 4.2.664). A Statistical_dimensional_tolerance may specify another Dimensional_characteristic as an alternate to the Statistical_dimensional_tolerance in the event that the specified level of process control cannot be met.

The data associated with a Statistical_dimensional_tolerance are the following:

- alternate_dimension;
- statistical_control_requirement.

NOTE Statistical tolerancing is discussed in 2.16 of ASME Y14.5M.

4.2.840.1 alternate_dimension

The alternate_dimension specifies the Dimensional_characteristic (see 4.2.298) that shall be used if the specified level of process control cannot be met. The alternate_dimension need not be specified for a Statistical_dimensional_tolerance.

4.2.840.2 statistical_control_requirement

The statistical_control_requirement specifies the Ee_requirement_occurrence (see 4.2.330) that states the statistical process control requirements that the manufacturing facility shall meet for the Statistical_dimensional_tolerance.

Constraints:

WR1: If a Dimensional_characteristic is specified as an alternate_dimension, the specified Dimensional_characteristic shall not be a Statistical_dimensional_tolerance.

WR2: If a Dimensional_characteristic is specified as an alternate_dimension, the Dimensional_characteristic shall characterize the same Shape_element or Shape_element_locating_relationship as characterized by the Statistical_dimensional_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY statistical_dimensional_tolerance
  SUBTYPE OF (dimensional_characteristic);
    alternate_dimension          : OPTIONAL dimensional_characteristic;
    statistical_control_requirement : ee_requirement_occurrence;
END_ENTITY; -- statistical_dimensional_tolerance

( *
```

4.2.841 Statistical_geometric_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Statistical_geometric_tolerance is a type of Geometric_tolerance (see 4.2.419) that specifies design requirements that are based on sound statistics. Associated with the Statistical_geometric_tolerance is the requirement that a specified level of process control be maintained during the manufacture of the Physical_unit (see 4.2.664). A Statistical_geometric_tolerance may specify another Geometric_tolerance as an alternate to the Statistical_geometric_tolerance in the event that the specified level of process control cannot be met.

The data associated with a Statistical_geometric_tolerance are the following:

- alternate_tolerance;
- statistical_control_requirement.

NOTE Statistical tolerancing is discussed in 2.16 of ASME Y14.5M.

4.2.841.1 alternate_tolerance

The alternate_tolerance specifies the Geometric_tolerance (see 4.2.419) that shall be used if the specified level of process control cannot be met. The alternate_tolerance need not be specified for a Statistical_geometric_tolerance.

4.2.841.2 statistical_control_requirement

The statistical_control_requirement specifies the Ee_requirement_occurrence (see 4.2.330) that states the statistical process control requirements that the manufacturing facility shall meet for the Statistical_geometric_tolerance.

Constraints:

WR1: If a Geometric_tolerance is specified as an alternate_tolerance, the specified Geometric_tolerance shall not be a Statistical_geometric_tolerance.

WR2: If a Geometric_tolerance is specified as an alternate_tolerance, the Shape_element specified as the tolerated_shape_element by the Statistical_geometric_tolerance shall be the same instance of Shape_element as specified as the tolerated_shape_element by the Geometric_tolerance specified as the alternate_tolerance.

NOTE The attribute tolerated_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

NOTE EXPRESS specification:

```

*)
ENTITY statistical_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
    alternate_tolerance          : OPTIONAL geometric_tolerance;
    statistical_control_requirement : ee_requirement_occurrence;
END_ENTITY; -- statistical_geometric_tolerance

( *
```

4.2.842 Statistical_tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Statistical_tolerance_characteristic is a type of Tolerance_characteristic (see 4.2.911). The Statistical_tolerance_characteristic supports the exchange of properties defined by statistical distributions.

Constraints:

IP1: The component_characteristic inherited from Coordinated_characteristic shall all participate in the characterization of one statistical tolerance.

IP2: The type_name inherited from Coordinated_characteristic shall be a recognized statistical tolerance type.

NOTE EXPRESS specification:

```
*)
ENTITY statistical_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
END_ENTITY; -- statistical_tolerance_characteristic

( *
```

4.2.843 Step_edge_feature_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Step_edge_feature_shape is a type of Edge_shape (see 4.2.313) that is the two-dimensional definition of the construction of a step edge. See Bevel_edge_feature_shape (see 4.2.69) for additional supporting geometric information that shall be provided.

EXAMPLE 1 Figure 102 illustrates a detail of a fabrication drawing that shows some of the information exchanged for construction of a step edge.

The data associated with a Step_edge_feature_shape are the following:

- maximum_step_depth;
- maximum_step_width;
- minimum_step_depth;
- minimum_step_width.

4.2.843.1 maximum_step_depth

The maximum_step_depth specifies the Length_data_element (see 4.2.511) of the maximum depth of the step relative to the datum surface.

4.2.843.2 maximum_step_width

The maximum_step_width specifies the Length_data_element (see 4.2.511) of the maximum width of the step relative to the edge surface that is stepped.

4.2.843.3 minimum_step_depth

The minimum_step_depth specifies the Length_data_element (see 4.2.511) of the minimum depth of the step relative to the datum surface.

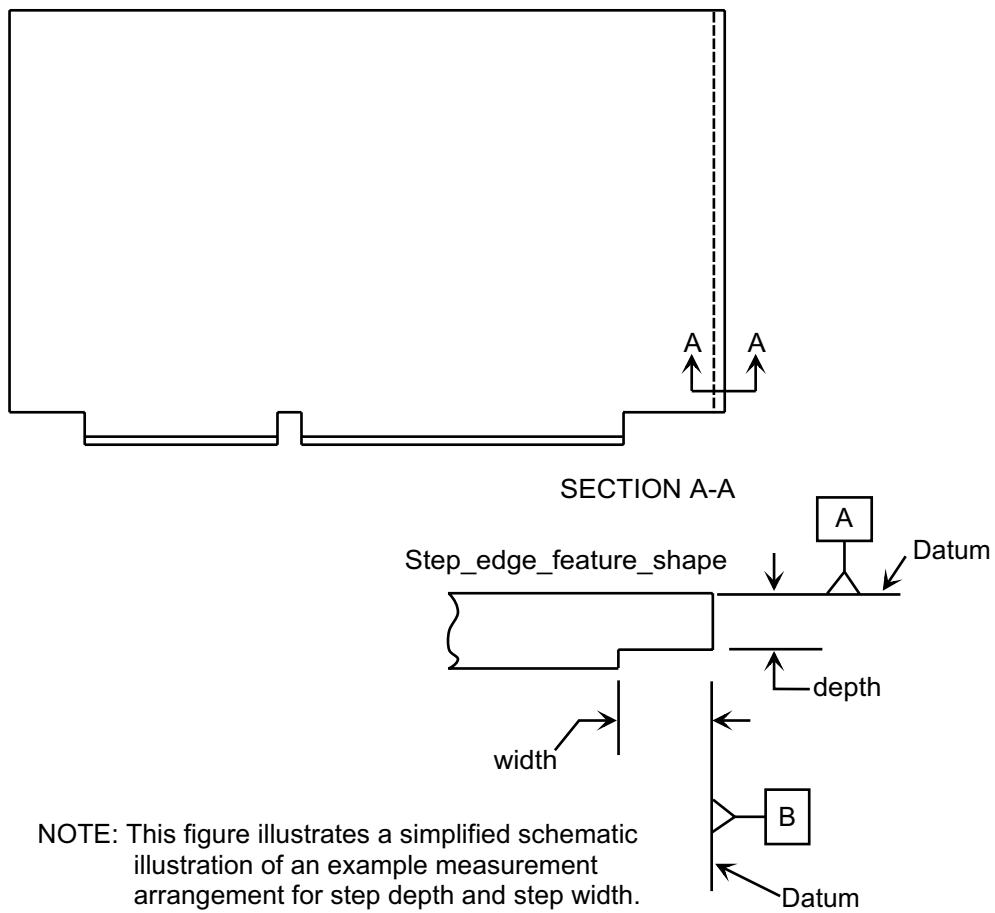


Figure 102 – Step_edge_feature_shape

4.2.843.4 minimum_step_width

The minimum_step_width specifies the Length_data_element (see 4.2.511) of the minimum width of the step relative to the edge surface that is stepped.

NOTE EXPRESS specification:

```

*)
ENTITY step_edge_feature_shape
  SUBTYPE OF (edge_shape);
    minimum_step_width : length_data_element;
    maximum_step_width : length_data_element;
    maximum_step_depth : length_data_element;
    minimum_step_depth : length_data_element;
END_ENTITY; -- step_edge_feature_shape

( *
```

4.2.844 Straightness_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Straightness_tolerance is a type of Geometric_tolerance_without_referenced_datum_system (see 4.2.423) that specifies or partially specifies the design requirement on the straightness of either the Shape_element (see 4.2.810) to which the Straightness_tolerance is applied or of constituent elements of the Shape_element to which the Straightness_tolerance is applied. When the design requirements are on the straightness of constituent elements of a non-axial Shape_element, the orientation of the Tolerance_zone_boundary (see 4.2.915) is specified by a Viewing_plane (see 4.2.956) to which the Tolerance_zone_boundary is parallel.

NOTE Straightness tolerance is defined in terms of tolerance zone definitions in 14.1 of ISO 1101.

Constraints:

WR1: The Shape_element specified by the toleranced_shape_element shall be either a Physical_feature_or_part_template, a Centre_axis or a Centre_plane

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: If the toleranced_shape_element is a Centre_axis, the Straightness_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify a Diametrical_boundary_size_characteristic as its specified_boundary_size; otherwise, the Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as the specified_boundary_size.

WR3: If the toleranced_shape_element is a Physical_feature_or_part_template, the Tolerance_zone_definition that is specified as the tolerance_zone_or_boundary_definition shall be a Tolerance_zone_definition_with_specified_orientation that specifies a Viewing_plane_based_orientation as its zone_orientation; otherwise, the Tolerance_zone_definition shall not be a Tolerance_zone_definition_with_specified_orientation.

IP1: The form of the tolerated Shape_element shall be consistent with the form of a shape element for which the application of a straightness tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY straightness_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- straightness_tolerance

( *
```

4.2.845 Stratum

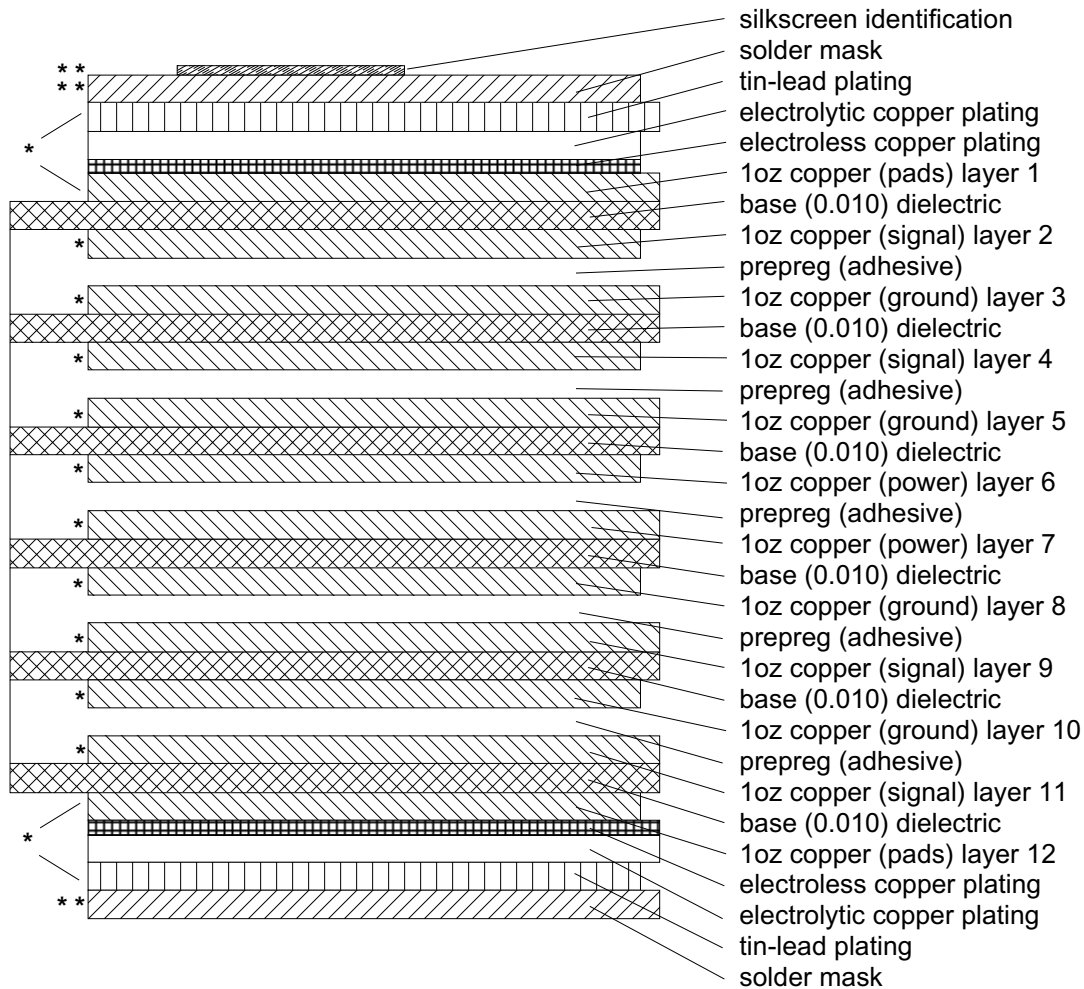
NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum is a type of Managed_design_object (see 4.2.524). A Stratum may be a Documentation_layer_stratum (see 4.2.306) or a Design_layer_stratum (see 4.2.275). A Stratum consists of a single material within an interconnect substrate. The vertical contours of either boundary surface may vary. Thus, the thickness of the stratum may vary throughout its range. The material used may be specific, such as 24K gold, or more general, such as etched copper. There may only be one boundary.

EXAMPLE 1 Figure 103 illustrates a detail of a stackup drawing included on a fabrication drawing in some cases. In most cases, the detail in this figure is generated at the fabricator's facility from more generic engineering data. This part of ISO 10303 is capable of supporting exchange of generic material stackup data or very explicit material stackup data. This part of ISO 10303 considers either stackup data set as design data. An enterprise may track the source of the stackup data through the design_supplier attribute of Ee_product_version (see 4.2.328).

The data associated with a Stratum are the following:

- assembly;
- bound_volume_shape;
- name;
- of_average_surface;
- of_surface;
- of_technology;
- planar_shape;
- stratum_usage.



```
*Design_layer_stratum  **Documentation_layer_stratum
```

This illustration shows different types of Stratum in a typical multi-layer printed circuit board cross section view. Stratum without footnote references are neither Design_layer_stratum or Documentation_layer_stratum.

Figure 103 – Stratum

4.2.845.1 assembly

The assembly is an inverse relationship that specifies the Interconnect_module_stratum_assembly_relationship (see 4.2.460) that specifies the Stratum as its component. There shall be one or more Interconnect_module_stratum_assembly_relationship for a Stratum, but they shall all be for the same product.

4.2.845.2 bound_volume_shape

The bound_volume_shape is an inverse relationship that specifies the Stratum_3d_shape (see 4.2.846) that specifies the Stratum as its shape_characterized_stratum. There shall be no more than one Stratum_3d_shape for a Stratum.

4.2.845.3 name

The name specifies an identifier of a Stratum and is equal to the product_definition_id attribute inherited from Ee_product_definition.

4.2.845.4 of_average_surface

An of_average_surface is an inverse attribute which specifies the Stratum_average_surface (see 4.2.847) of the Stratum as its of_stratum. There shall be no more than one Stratum_average_surface for a Stratum.

4.2.845.5 of_surface

An of_surface is an inverse attribute which specifies one or two Stratum_surface (see 4.2.868) for a Stratum.

4.2.845.6 of_technology

The of_technology specifies the role of the Stratum_technology (see 4.2.871) for the Stratum.

4.2.845.7 planar_shape

The planar_shape is an inverse relationship that specifies the Stratum_planar_shape (see 4.2.867) that specifies the Stratum as its shape_characterized_stratum. There shall be no more than one Stratum_planar_shape for a Stratum.

4.2.845.8 stratum_usage

The stratum_usage specifies the External_definition (see 4.2.361) that describes the purpose for incorporating a stratum in the Interconnect_module (see 4.2.440) design. The stratum_usage need not be specified for a particular Stratum.

NOTE EXPRESS specification:

```

*)
ENTITY stratum
  SUPERTYPE OF (ONEOF (documentation_layer_stratum, design_layer_stratum))
  SUBTYPE OF (ee_product_definition);
  stratum_usage : OPTIONAL external_definition;
  of_technology : stratum_technology;
DERIVE
  name          : ee_name := SELF\ee_product_definition.product_definition_id;
INVERSE
  of_average_surface : SET [0:1] OF stratum_average_surface FOR of_stratum;
  of_surface         : SET [1:2] OF stratum_surface FOR of_stratum;
  planar_shape       : SET [0:1] OF stratum_planar_shape FOR
    shape_characterized_stratum;
  assembly           : SET [1:?] OF
    interconnect_module_stratum_assembly_relationship FOR
    component;
  bound_volume_shape : SET [0:1] OF stratum_3d_shape FOR
    shape_characterized_stratum;
END_ENTITY; -- stratum
( *

```

4.2.846 Stratum_3d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_3d_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Stratum_3d_shape is the three-dimensional shape of a Stratum (see 4.2.845).

The data associated with a Stratum_3d_shape are the following:

— shape_characterized_stratum.

4.2.846.1 shape_characterized_stratum

The shape_characterized_stratum specifies the role of the Stratum (see 4.2.845) for the Stratum_3d_shape.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_3d_shape
  SUBTYPE OF (bound_volume_shape, feature_shape_definition);
  shape_characterized_stratum : stratum;
END_ENTITY; -- stratum_3d_shape
( *

```

4.2.847 Stratum_average_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_average_surface is a type of Managed_design_object (see 4.2.524). A Stratum_average_surface specifies the constructed geometry that is the average of the primary and secondary surface for a Stratum (see 4.2.845). A Stratum_average_surface is provided as a construction aid in defining a Stratum.

The data associated with a Stratum_average_surface are the following:

- of_shape;
- of_stratum.

4.2.847.1 of_shape

An of_shape is an inverse relationship that specifies the role of a Stratum_average_surface_shape (see 4.2.848) for a Stratum_average_surface. An of_shape need not be specified for a Stratum_average_surface.

4.2.847.2 of_stratum

An of_stratum specifies the role of a Stratum (see 4.2.845) for a Stratum_average_surface.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_average_surface
  SUBTYPE OF (managed_design_object);
    of_stratum : stratum;
  INVERSE
    of_shape : SET [0:1] OF stratum_average_surface_shape FOR of_surface;
END_ENTITY; -- stratum_average_surface

```

(*

4.2.848 Stratum_average_surface_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_average_surface_shape is a type of Open_shell_based_surface (see 4.2.566). A Stratum_average_surface_shape is the representation of a Stratum_average_surface (see 4.2.847) using an open manifold surface.

The data associated with a Stratum_average_surface_shape are the following:

— of_surface.

4.2.848.1 of_surface

The of_surface specifies the role of a Stratum_average_surface (see 4.2.847) for the Stratum_average_surface_shape.

NOTE EXPRESS specification:

```
* )
ENTITY stratum_average_surface_shape
  SUBTYPE OF (open_shell_based_surface);
  of_surface : stratum_average_surface;
END_ENTITY; -- stratum_average_surface_shape
```

(*

4.2.849 Stratum_concept_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_concept_relationship is a type of Managed_design_object (see 4.2.524). A Stratum_concept_relationship is a Dielectric_crossover_area (see 4.2.288) or a Stratum_feature_conductive_join (see 4.2.853).

The data associated with a Stratum_concept_relationship are the following:

— resulting_shape;

— technology_requirement.

4.2.849.1 resulting_shape

The resulting_shape specifies the role of the Feature_shape_definition (see 4.2.372) for the Stratum_concept_relationship. The resulting_shape need not be specified for a particular Stratum_concept_relationship.

4.2.849.2 technology_requirement

The technology_requirement specifies the Ee_requirement_occurrence (see 4.2.330) that results in the resulting_shape. There shall be one or more Ee_requirement_occurrence for a Stratum_concept_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_concept_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (stratum_feature_conductive_join,
    dielectric_crossover_area))
  SUBTYPE OF (managed_design_object);
    resulting_shape      : OPTIONAL feature_shape_definition;
    technology_requirement : SET [1:?] OF ee_requirement_occurrence;
END_ENTITY; -- stratum_concept_relationship

( *

```

4.2.850 Stratum_concept_to_physical_usage_view_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_concept_to_physical_usage_view_assignment is a type of Managed_design_object (see 4.2.524). A Stratum_concept_to_physical_usage_view_assignment is the relationship between a Part_feature (see 4.2.609) and any one of an Inter_stratum_feature (see 4.2.432), a Laminate_component (see 4.2.483), a Stratum (see 4.2.845), Stratum_feature (see 4.2.851), or a Stratum_surface (see 4.2.868) for the purpose of traceability between the Part_feature and the underlying design feature.

The data associated with a Stratum_concept_to_physical_usage_view_assignment are the following:

- assigned_design_object;
- defined_part_feature.

4.2.850.1 assigned_design_object

The assigned_design_object specifies any one of the Inter_stratum_feature (see 4.2.432), Laminate_component (see 4.2.483), Stratum (see 4.2.845), Stratum_feature (see 4.2.851), or Stratum_surface (see 4.2.868) as the assigned object for the Stratum_concept_to_physical_usage_view_assignment.

4.2.850.2 defined_part_feature

The defined_part_feature specifies the role of the Part_feature (see 4.2.609) for the Stratum_concept_to_physical_usage_view_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_concept_to_physical_usage_view_assignment
  SUBTYPE OF (managed_design_object);
  assigned_design_object : stratum_concept;
  defined_part_feature   : part_feature;
WHERE
  WR1: NOT ( 'AP210_ARM.INTERCONNECT_MODULE_TERMINAL'
             IN TYPEOF(defined_part_feature));
END_ENTITY; -- stratum_concept_to_physical_usage_view_assignment

( *

```

4.2.851 Stratum_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature is a type of Physical_feature_or_part_template (see 4.2.658). A Stratum_feature may be either a Conductive_filled_area (see 4.2.170) or a Conductor (see 4.2.176), or additionally may be a Fiducial_stratum_feature (see 4.2.375). A Stratum_feature is a shape realized in the material of the Stratum (see 4.2.845).

EXAMPLE 1 Figure 104 illustrates several types of Stratum_feature. The Conductors shown are purposely distorted in aspect ratio because Conductors are enclosed shapes.

The data associated with a Stratum_feature are the following:

- feature_of_size;
- name;
- resident_stratum.

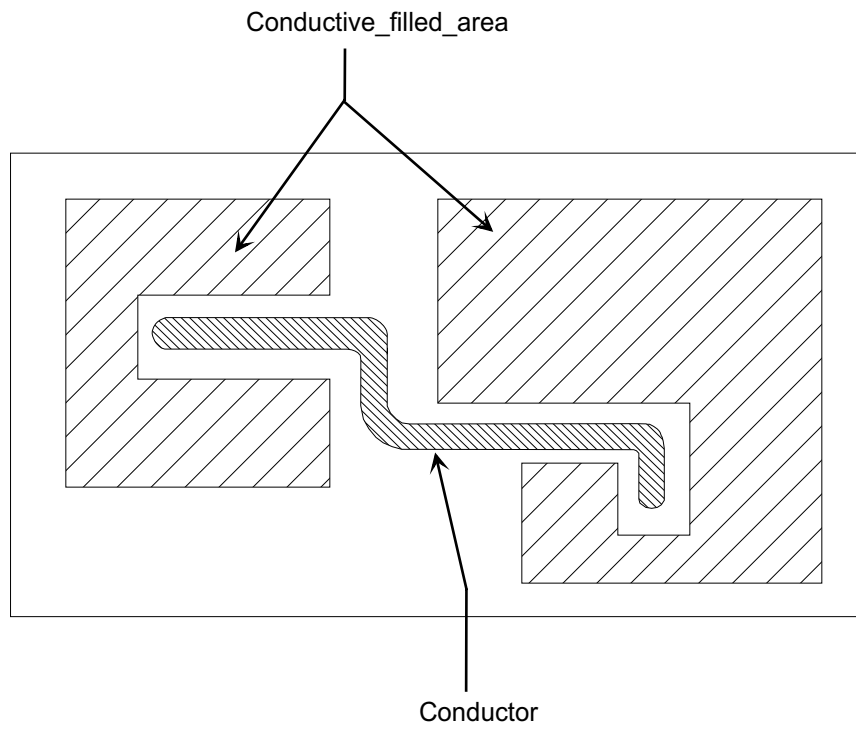
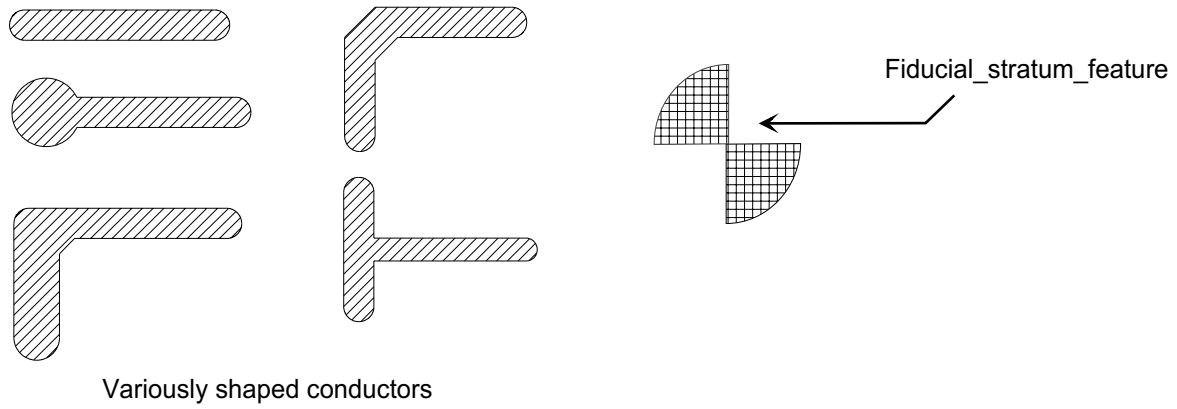
4.2.851.1 feature_of_size

The feature_of_size specifies whether or not the Stratum_feature interfaces with a feature of another product in an assembly context. A value of TRUE indicates that the Stratum_feature interfaces with a feature of another product. A value of FALSE indicates that the Stratum_feature does not interface with a feature of another product.

NOTE Certain tolerance information depends on knowledge of a particular feature interfacing with features of a mating product.

4.2.851.2 name

The name specifies the identifier of the Stratum_feature.



Stratum_feature types

Figure 104 – Stratum_feature

4.2.851.3 resident_stratum

The resident_stratum specifies the Stratum (see 4.2.845) being composed by the Stratum_feature.

Constraints:

UR1: The combination of the name and the resident_stratum shall be unique within a population of Stratum_features.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature
  SUPERTYPE OF (ONEOF (conductor,conductive_filled_area) ANDOR
    fiducial_stratum_feature)
  SUBTYPE OF (physical_feature_or_part_template);
  resident_stratum      : stratum;
  SELF\shape_element.name : ee_name;
  feature_of_size       : BOOLEAN;
  UNIQUE
    UR1 : name, resident_stratum;
END_ENTITY; -- stratum_feature

```

(*

4.2.852 Stratum_feature_3d_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature_3d_shape is a type of Bound_volume_shape (see 4.2.76) and a type of Feature_shape_definition (see 4.2.372). A Stratum_feature_3d_shape is the representation of a Stratum_feature (see 4.2.851) using a closed volume.

The data associated with a Stratum_feature_3d_shape are the following:

- represented_stratum_feature;
- shape_representation.

4.2.852.1 represented_stratum_feature

The represented_stratum_feature specifies the role of a Stratum_feature (see 4.2.851) for the Stratum_feature_3d_shape.

4.2.852.2 shape_representation

The shape_representation specifies the role of a Bound_volume_shape (see 4.2.76) for the Stratum_feature_3d_shape.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_3d_shape
  SUBTYPE OF (bound_volume_shape, feature_shape_definition);
    represented_stratum_feature : stratum_feature;
(* deleted tt tc1 wd9
    shape_surface_representation : bound_volume_shape;
*)
END_ENTITY; -- stratum_feature_3d_shape

--tt tc1 wd9 change
( *
```

4.2.853 Stratum_feature_conductive_join

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature_conductive_join is a type of Stratum_concept_relationship (see 4.2.849). A Stratum_feature_conductive_join may be a Physical_network_supporting_stratum_feature_conductive_join (see 4.2.662). A Stratum_feature_conductive_join is a connection between two conductive Stratum_feature (see 4.2.851) on disjoint Stratum (see 4.2.845) that is not a passage.

NOTE The two facing surfaces of conductive features are joined.

The data associated with a Stratum_feature_conductive_join are the following:

- related_concept;
- relating_concept;
- using_via.

4.2.853.1 related_concept

The related_concept specifies the source Stratum_feature_template_component (see 4.2.862) for the Stratum_feature_conductive_join.

4.2.853.2 relating_concept

The relating_concept specifies the target Stratum_feature_template_component (see 4.2.862) for the Stratum_feature_conductive_join.

4.2.853.3 using_via

The using_via is an inverse relationship that specifies that there shall be no more than one Bonded_conductive_base_blind_via (see 4.2.74) that specifies the Stratum_feature_conductive_join as its features_join. The using_via need not be specified for a particular Stratum_feature_conductive_join.

Constraints:

WR1: Each member of Stratum_feature_template_component in the role of relating_concept for this Stratum_feature_conductive_join shall not be in the role of related_concept for this member of Stratum_feature_conductive_join.

WR2: A Stratum_feature_template_component shall not be an ancestor or descendant of itself in this relationship.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_conductive_join
  SUPERTYPE OF (
    physical_network_supporting_stratum_feature_conductive_join)
  SUBTYPE OF (stratum_concept_relationship);
  related_concept : stratum_feature_template_component;
  relating_concept : stratum_feature_template_component;
  INVERSE
    using_via : SET [0:1] OF bonded_conductive_base_blind_via FOR features_join;
WHERE
  WR1: related_concept.stratum_feature_implementation.resident_stratum :&#60;&#62;;
    relating_concept.stratum_feature_implementation.resident_stratum;
END_ENTITY; -- stratum_feature_conductive_join

( *
```

4.2.854 Stratum_feature_in_stratum_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature_in_stratum_3d_position is a type of Managed_design_object (see 4.2.524). A Stratum_feature_in_stratum_3d_position is the three-dimensional position and orientation of a three-dimensional model of a Stratum_feature (see 4.2.851) in the three-dimensional model of the Stratum (see 4.2.845).

The data associated with a Stratum_feature_in_stratum_3d_position are the following:

- feature_shape;
- stratum_shape;
- transformation.

4.2.854.1 feature_shape

The feature_shape specifies the role of either the Stratum_feature_3d_shape (see 4.2.852), or the Stratum_feature_non_planar_2d_shape (see 4.2.856) for the Stratum_feature_in_stratum_3d_position.

4.2.854.2 stratum_shape

The stratum_shape specifies the role of the Stratum_3d_shape (see 4.2.846) for the Stratum_feature_in_stratum_3d_position.

4.2.854.3 transformation

The transformation specifies the role of the Axis_placement_3d (see 4.2.58) for the Stratum_feature_in_stratum_3d_position.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_in_stratum_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_3d_shape;
    transformation : axis_placement_3d;
    feature_shape : stratum_feature_3d_shape_select;
END_ENTITY; -- stratum_feature_in_stratum_3d_position

( *
```

4.2.855 Stratum_feature_in_stratum_planar_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature_in_stratum_planar_position is a type of Managed_design_object (see 4.2.524). A Stratum_feature_in_stratum_planar_position is the two-dimensional position and orientation of a two-dimensional model of a Stratum_feature (see 4.2.851) in the two-dimensional model of the Stratum (see 4.2.845).

The data associated with a `Stratum_feature_in_stratum_planar_position` are the following:

- `feature_shape`;
- `stratum_shape`;
- `transformation`.

4.2.855.1 `feature_shape`

The `feature_shape` specifies either a `Stratum_feature_planar_shape` (see 4.2.857) or a `Stratum_feature_styled_area_representation` (see 4.2.860) for the `Stratum_feature_in_stratum_planar_position`.

4.2.855.2 `stratum_shape`

The `stratum_shape` specifies the role of the `Stratum_planar_shape` (see 4.2.867) for the `Stratum_feature_in_stratum_planar_position`.

4.2.855.3 `transformation`

The `transformation` specifies the role of the `Axis_placement_2d` (see 4.2.57) for the `Stratum_feature_in_stratum_planar_position`.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_in_stratum_planar_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_planar_shape;
    transformation : axis_placement_2d;
    feature_shape : stratum_feature_planar_representation_select;
END_ENTITY; -- stratum_feature_in_stratum_planar_position

( *
```

4.2.856 `Stratum_feature_non_planar_2d_shape`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_feature_non_planar_2d_shape` is a type of `Managed_design_object` (see 4.2.524). A `Stratum_feature_non_planar_2d_shape` is the representation of a `Stratum_feature` (see 4.2.851) using an open manifold surface.

The data associated with a `Stratum_feature_non_planar_2d_shape` are the following:

- `represented_stratum_feature`;
- `shape_representation`.

4.2.856.1 `represented_stratum_feature`

The `represented_stratum_feature` specifies the role of a `Stratum_feature` (see 4.2.851) for the `Stratum_feature_non_planar_2d_shape`.

4.2.856.2 `shape_representation`

The `shape_representation` specifies the role of a `Open_shell_based_surface` (see 4.2.566) for the `Stratum_feature_non_planar_2d_shape`.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_non_planar_2d_shape
--tt tc1 wd9 modified
    SUBTYPE OF (open_shell_based_surface);
(* tt tc1 wd9 deleted
    shape_representation          : open_shell_based_surface;
*)
    represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_non_planar_2d_shape

( *
```

4.2.857 `Stratum_feature_planar_shape`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_feature_planar_shape` is a type of `Managed_design_object` (see 4.2.524). A `Stratum_feature_planar_shape` is the representation of a `Stratum_feature` (see 4.2.851) by a `Planar_projected_shape` (see 4.2.677).

The data associated with a `Stratum_feature_planar_shape` are the following:

- `represented_stratum_feature`;
- `shape_representation`.

4.2.857.1 represented_stratum_feature

The `represented_stratum_feature` specifies the `Stratum_feature` (see 4.2.851) that is to be characterized by the `Stratum_feature_planar_shape`.

4.2.857.2 shape_representation

The `shape_representation` specifies the `Planar_projected_shape` (see 4.2.677) for the representation of the `Stratum_feature` (see 4.2.851).

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_planar_shape
  SUBTYPE OF (planar_projected_shape);
(* tt tc1 wd9 deleted
  shape_representation          : planar_projected_shape;
*)
  represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_planar_shape

( *
```

4.2.858 Stratum_feature_shape_stratum_average_surface_shape_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_feature_shape_stratum_average_surface_shape_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Stratum_feature_shape_stratum_average_surface_shape_relationship` is the association between a `Stratum_feature_non_planar_2d_shape` (see 4.2.856) and a `Stratum_average_surface_shape` (see 4.2.848) for the purpose of specifying that the points on the `Stratum_feature_non_planar_2d_shape` shall be in the domain of the points on the `Stratum_average_surface_shape`.

The data associated with a `Stratum_feature_shape_stratum_average_surface_shape_relationship` are the following:

- `of_stratum_feature_shape`;
- `of_stratum_average_surface_shape`.

4.2.858.1 of_stratum_feature_shape

The `of_stratum_feature_shape` specifies the dependent `Stratum_feature_non_planar_2d_shape` (see 4.2.856) for the `Stratum_feature_shape_stratum_average_surface_shape_relationship`.

4.2.858.2 of_stratum_average_surface_shape

The of_stratum_average_surface_shape specifies the independent Stratum_average_surface_shape (see 4.2.848) for the Stratum_feature_shape_stratum_average_surface_shape_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_shape_stratum_average_surface_shape_relationship
  SUBTYPE OF (managed_design_object);
    of_stratum_feature_shape      : stratum_feature_non_planar_2d_shape;
    of_stratum_average_surface_shape : stratum_average_surface_shape;
END_ENTITY; -- stratum_feature_shape_stratum_average_surface_shape_relationship

( *
```

4.2.859 Stratum_feature_shape_stratum_surface_shape_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_feature_shape_stratum_surface_shape_relationship is a type of Managed_design_object (see 4.2.524). A Stratum_feature_shape_stratum_surface_shape_relationship is the association between a Stratum_feature_non_planar_2d_shape (see 4.2.856) and a Stratum_surface_shape (see 4.2.870) for the purpose of specifying that the points on the Stratum_feature_non_planar_2d_shape shall be in the domain of the points on the Stratum_surface_shape.

The data associated with a Stratum_feature_shape_stratum_surface_shape_relationship are the following:

- of_stratum_feature_shape;
- of_stratum_surface_shape.

4.2.859.1 of_stratum_feature_shape

The of_stratum_feature_shape specifies the dependent Stratum_feature_non_planar_2d_shape (see 4.2.856) for the Stratum_feature_shape_stratum_surface_shape_relationship.

4.2.859.2 of_stratum_surface_shape

The of_stratum_surface_shape specifies the independent Stratum_surface_shape (see 4.2.870) for the Stratum_feature_shape_stratum_surface_shape_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_shape_stratum_surface_shape_relationship
  SUBTYPE OF (managed_design_object);
    of_stratum_feature_shape : stratum_feature_non_planar_2d_shape;
    of_stratum_surface_shape : stratum_surface_shape;
END_ENTITY; -- stratum_feature_shape_stratum_surface_shape_relationship

( *

```

4.2.860 Stratum_feature_styled_area_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_feature_styled_area_representation` is a type of `Managed_design_object` (see 4.2.524). A `Stratum_feature_styled_area_representation` is the association between one or more of a `Stratum_feature` (see 4.2.851) and one of a `Fill_area_occurrence` (see 4.2.377), or a `Styled_curve_occurrence` (see 4.2.875). The `Stratum_feature_styled_area_representation` is intended for use where legacy data may need to be exchanged via this Standard or where complex patterns of `Stratum_feature` are to be created. The `Stratum_feature_styled_area_representation` specifies parametric data to be input to an algorithm. The algorithm output shall be the individual `Stratum_feature` shapes. Once the individual `Stratum_feature` shapes are available, they may be used if explicit shape data is required by a receiving processor. If the individual shapes are provided, the attribute `derived_planar_shape` shall be populated to identify the derived shapes. The units used in the `Stratum_feature_styled_area_representation` shall be those specified in the global geometric context for the `Interconnect_module` (see 4.2.440) design as specified by `Cartesian_coordinate_system` (see 4.2.86).

NOTE A `Stratum_feature_styled_area_representation` is not intended to be used to represent a complex `Stratum_feature` resulting from multiple design intent modifications to a basic shape. The other types of `Stratum_feature` shape representation should be used to convey that information.

EXAMPLE 1 The `Stratum_feature_styled_area_representation` for a "dot" pattern used in manufacturing will result in multiple `Stratum_feature` for one instance of `Stratum_feature_styled_area_representation`. Legacy vector formats that "paint" filled areas using a successive approximation application of a constant width curve may be supported with `Stratum_feature_styled_area_representation` using the `Styled_curve_occurrence` option. If the curve width varies, use of multiple `Design_intent_modification_styled_area_representation` (see 4.2.274) may be appropriate.

The data associated with a `Stratum_feature_styled_area_representation` are the following:

- `derived_planar_shape`;
- `represented_stratum_feature`;
- `shape_styled_area_representation`.

4.2.860.1 derived_planar_shape

The `derived_planar_shape` specifies the role of the `Stratum_feature_planar_shape` (see 4.2.857) for the `Stratum_feature_styled_area_representation`. There may be more than one `Stratum_feature_planar_shape` for the `Stratum_feature_styled_area_representation` if the `Stratum_feature_styled_area_representation` is discontinuous. The `Stratum_feature_planar_shape` need not be supplied for the `Stratum_feature_styled_area_representation`.

4.2.860.2 represented_stratum_feature

The `represented_stratum_feature` specifies the role of the `Stratum_feature` (see 4.2.851) for the `Stratum_feature_styled_area_representation`. There may be more than one `Stratum_feature` for the `Stratum_feature_styled_area_representation` if the `Stratum_feature_styled_area_representation` is discontinuous.

NOTE More than one `Stratum_feature` is required if the shape of the `Stratum_feature_styled_area_representation` is not a contiguous shape, since `Stratum_feature` is required to be a single contiguous shape.

4.2.860.3 shape_styled_area_representation

The `shape_styled_area_representation` specifies one of a `Fill_area_occurrence` (see 4.2.377), or a `Styled_curve_occurrence` (see 4.2.875) as the representation for the `Stratum_feature_styled_area_representation`.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_feature_styled_area_representation
  SUBTYPE OF (managed_design_object);
    derived_planar_shape          : OPTIONAL SET [1:?] OF
                                   stratum_feature_planar_shape;
    shape_styled_area_representation : styled_area_representation;
    represented_stratum_feature    : stratum_feature;
END_ENTITY; -- stratum_feature_styled_area_representation

( *
```

4.2.861 Stratum_feature_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_feature_template` is a type of `Part_template` (see 4.2.620) that is a definition or partial definition of a `Stratum_feature` (see 4.2.851). A `Stratum_feature_template` may be a `Land_physical_template` (see 4.2.493). A `Stratum_feature_template` defines a contiguous material shape.

The data associated with a `Stratum_feature_template` are the following:

— `of_stratum_technology`.

4.2.861.1 of_stratum_technology

The of_stratum_technology specifies the role of the Stratum_technology (see 4.2.871) for the Material_removal_feature_template.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_template
  SUPERTYPE OF (land_physical_template)
  SUBTYPE OF (part_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- stratum_feature_template

( *
```

4.2.862 Stratum_feature_template_component

A Stratum_feature_template_component is a type of Laminate_component (see 4.2.483). A Stratum_feature_template_component may be a Land (see 4.2.490) or a Non_functional_land (see 4.2.562).

The data associated with a Stratum_feature_template_component are the following:

- stratum_feature_implementation;
- usage_definition.

4.2.862.1 stratum_feature_implementation

The stratum_feature_implementation specifies the Stratum_feature (see 4.2.851) that the Stratum_feature_template_component is helping to compose.

4.2.862.2 usage_definition

The usage_definition specifies the role of the Stratum_feature_template (see 4.2.861) for the Stratum_feature_template_component.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_feature_template_component
  SUPERTYPE OF (ONEOF (connected_area_component,
                        non_functional_land, land))
  SUBTYPE OF (laminate_component);
  SELF\assembly_component.usage_definition : OPTIONAL stratum_feature_template;
  stratum_feature_implementation           : stratum_feature;
END_ENTITY; -- stratum_feature_template_component
( *

```

4.2.863 Stratum_interconnect_module_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_interconnect_module_3d_position is a type of Managed_design_object (see 4.2.524). A Stratum_interconnect_module_3d_position is the position of the Stratum (see 4.2.845) model in the Interconnect_module (see 4.2.440) model in three dimensions.

The data associated with a Stratum_interconnect_module_3d_position are the following:

- interconnect_module_shape;
- stratum_component;
- stratum_shape;
- transformation.

4.2.863.1 interconnect_module_shape

The interconnect_module_shape specifies the Physical_unit_3d_shape (see 4.2.666) that is the Interconnect_module (see 4.2.440) model for the Stratum_interconnect_module_3d_position.

4.2.863.2 stratum_component

The stratum_component specifies the role of the Interconnect_module_stratum_assembly_relationship (see 4.2.460) for the Stratum_interconnect_module_3d_position.

4.2.863.3 stratum_shape

The stratum_shape specifies the Stratum_3d_shape (see 4.2.846) for the Stratum (see 4.2.845) model for the Stratum_interconnect_module_3d_position.

4.2.863.4 transformation

The transformation specifies the role of the `Axis_placement_3d` (see 4.2.58) for the `Stratum_interconnect_module_3d_position`.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_interconnect_module_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_component      : interconnect_module_stratum_assembly_relationship;
    stratum_shape          : stratum_3d_shape;
    interconnect_module_shape : physical_unit_3d_shape;
    transformation         : axis_placement_3d;
END_ENTITY; -- stratum_interconnect_module_3d_position

( *
```

4.2.864 Stratum_interconnect_module_planar_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Stratum_interconnect_module_planar_position` is a type of `Managed_design_object` (see 4.2.524). A `Stratum_interconnect_module_planar_position` is the position of the Stratum (see 4.2.845) model in the `Interconnect_module` (see 4.2.440) model in two dimensions.

The data associated with a `Stratum_interconnect_module_planar_position` are the following:

- `interconnect_module_shape`;
- `stratum_component`;
- `stratum_shape`;
- `transformation`.

4.2.864.1 interconnect_module_shape

The `interconnect_module_shape` specifies the `Physical_unit_planar_shape` (see 4.2.670) for the `Interconnect_module` (see 4.2.440) model for the `Stratum_interconnect_module_planar_position`.

4.2.864.2 stratum_component

The `stratum_component` specifies the role of the `Interconnect_module_stratum_assembly_relationship` (see 4.2.460) for the `Stratum_interconnect_module_planar_position`.

4.2.864.3 stratum_shape

The stratum_shape specifies the Stratum_planar_shape (see 4.2.867) for the Stratum (see 4.2.845) model for the Stratum_interconnect_module_planar_position.

4.2.864.4 transformation

The transformation specifies the role of the Axis_placement_2d (see 4.2.57) for the Stratum_interconnect_module_planar_position.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_interconnect_module_planar_position
  SUBTYPE OF (managed_design_object);
    stratum_component      : interconnect_module_stratum_assembly_relationship;
    stratum_shape          : stratum_planar_shape;
    interconnect_module_shape : physical_unit_planar_shape;
    transformation         : axis_placement_2d;
END_ENTITY; -- stratum_interconnect_module_planar_position

( *
```

4.2.865 Stratum_make_from_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_make_from_relationship is a type of Managed_design_object (see 4.2.524). A Stratum_make_from_relationship is an association between a Stratum (see 4.2.845) whose design is intended to be reused and a Stratum that results from the reuse. The resultant_stratum may differ from the reusable_stratum for the particular Interconnect_module (see 4.2.440) design.

The data associated with a Stratum_make_from_relationship are the following:

- resultant_stratum;
- reusable_stratum.

4.2.865.1 resultant_stratum

The resultant_stratum specifies the Stratum (see 4.2.845) for the result of a design activity that started with reusable_stratum.

4.2.865.2 reusable_stratum

The reusable_stratum specifies the Stratum (see 4.2.845) that is intended to be used as a reference from which to create a resultant_stratum.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_stratum : stratum;
    resultant_stratum : stratum;
END_ENTITY; -- stratum_make_from_relationship

( *
```

4.2.866 Stratum_occurrence_specific_padstack_definition

A Stratum_occurrence_specific_padstack_definition is a type of Padstack_definition (see 4.2.595).

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_occurrence_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
    reference_stratum : stratum;
END_ENTITY; -- stratum_occurrence_specific_padstack_definition

( *
```

4.2.867 Stratum_planar_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_planar_shape is a type of Managed_design_object (see 4.2.524). A Stratum_planar_shape is the two-dimensional shape model of a Stratum (see 4.2.845).

The data associated with a Stratum_planar_shape are the following:

— shape_characterized_stratum.

4.2.867.1 shape_characterized_stratum

The shape_characterized_stratum specifies the role of the Stratum (see 4.2.845) for the Stratum_planar_shape.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_planar_shape
--tt tc1 wd9 modified
  SUBTYPE OF (planar_projected_shape);
  shape_characterized_stratum : stratum;
END_ENTITY; -- stratum_planar_shape
```

(*

4.2.868 Stratum_surface

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_surface is a type of Managed_design_object (see 4.2.524). A Stratum_surface specifies the primary_surface or secondary_surface for a Stratum (see 4.2.845).

The data associated with a Stratum_surface are the following:

- name;
- of_stratum.

4.2.868.1 name

The name designates a primary_surface or a secondary_surface for a Stratum_surface.

4.2.868.2 of_stratum

An of_stratum specifies the Stratum (see 4.2.845) for a Stratum_surface.

Constraints:

UR1: The combination of name and of_stratum shall be unique within a population of Stratum_surface.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_surface
  SUBTYPE OF (managed_design_object);
    of_stratum : stratum;
    name       : stratum_surface_designation;
  INVERSE
    of_shape : SET [0:1] OF stratum_surface_shape FOR of_surface;
  UNIQUE
    UR1 : name, of_stratum;
END_ENTITY; -- stratum_surface

```

(*

4.2.869 Stratum_surface_in_stratum_3d_position

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_surface_in_stratum_3d_position is a type of Managed_design_object (see 4.2.524). A Stratum_surface_in_stratum_3d_position is the three-dimensional position and orientation of a two-dimensional constructed surface model in the three-dimensional model of the Stratum (see 4.2.845).

NOTE The surface represented by the construct may be constructed in a CAD system as a design abstraction or may be a realizeable surface.

The data associated with a Stratum_surface_in_stratum_3d_position are the following:

- stratum_shape;
- surface_shape;
- transformation.

4.2.869.1 stratum_shape

The stratum_shape specifies the role of the Stratum_3d_shape (see 4.2.846) for the Stratum_surface_in_stratum_3d_position.

4.2.869.2 surface_shape

The surface_shape specifies the role of either the Stratum_average_surface_shape (see 4.2.848) or of the Stratum_surface_shape (see 4.2.870) for the Stratum_surface_in_stratum_3d_position.

4.2.869.3 transformation

The transformation specifies the role of the Axis_placement_3d (see 4.2.58) for the Stratum_surface_in_stratum_3d_position.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_surface_in_stratum_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_3d_shape;
    transformation : axis_placement_3d;
    surface_shape : stratum_surface_shape_select;
END_ENTITY; -- stratum_surface_in_stratum_3d_position
```

(*

4.2.870 Stratum_surface_shape

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_surface_shape is a type of Open_shell_based_surface (see 4.2.566). A Stratum_surface_shape is the representation of a Stratum_surface (see 4.2.868) using an open manifold surface.

The data associated with a Stratum_surface_shape are the following:

— of_surface.

4.2.870.1 of_surface

The of_surface specifies the role of a Stratum_surface (see 4.2.868) for the Stratum_surface_shape.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_surface_shape
  SUBTYPE OF (open_shell_based_surface);
    of_surface : stratum_surface;
END_ENTITY; -- stratum_surface_shape
```

(*

4.2.871 Stratum_technology

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_technology is a type of Managed_design_object (see 4.2.524). A Stratum_technology may be either a Design_layer_technology (see 4.2.276) or a Documentation_layer_technology (see 4.2.307). A Stratum_technology provides material properties, parametric data associated with the technology, and material identification.

The data associated with a Stratum_technology are the following:

- laminate_stiffness_class;
- layer_position;
- maximum_feature_size_requirement;
- maximum_stratum_thickness;
- minimum_finished_feature_size;
- minimum_finished_feature_spacing;
- minimum_stratum_thickness;
- name;
- specification;
- stratum_material.

4.2.871.1 laminate_stiffness_class

The laminate_stiffness_class specifies either a conformal_coat, a fluid_like, or a stiff_laminate stiffness class of which the Stratum_technology is a member. The laminate_stiffness_class need not be specified for a particular Stratum_technology.

4.2.871.2 layer_position

The layer_position specifies one of an internal, external, all, primary, or secondary position of the Stratum (see 4.2.845) for the Stratum_technology. Primary and secondary shall only be used if the member of Stratum_technology is a Design_layer_technology.

4.2.871.3 maximum_feature_size_requirement

The maximum_feature_size_requirement specifies the maximum Length_data_element (see 4.2.511) of any feature of the Stratum (see 4.2.845). The maximum_feature_size_requirement need not be specified for a particular Stratum_technology.

4.2.871.4 maximum_stratum_thickness

The maximum_stratum_thickness specifies the maximum Length_data_element (see 4.2.511) for the thickness of the Stratum_technology.

4.2.871.5 minimum_finished_feature_size

The minimum_finished_feature_size specifies one role of the Length_data_element (see 4.2.511) of any feature of the Stratum (see 4.2.845). The minimum_finished_feature_size need not be specified for a particular Stratum_technology.

4.2.871.6 minimum_finished_feature_spacing

The minimum_finished_feature_spacing specifies one role of the Length_data_element (see 4.2.511) of any feature of the Stratum (see 4.2.845). The minimum_finished_feature_spacing need not be specified for a particular Stratum_technology.

4.2.871.7 minimum_stratum_thickness

The minimum_stratum_thickness specifies the minimum Length_data_element (see 4.2.511) for the thickness of the Stratum_technology.

4.2.871.8 name

The name specifies the alphanumeric identifier of a Stratum_technology.

4.2.871.9 specification

The specification specifies the role of the Process_specification (see 4.2.740) for the Stratum_technology. The specification need not be specified for a particular Stratum_technology.

4.2.871.10 stratum_material

The stratum_material specifies the role of the Ee_material (see 4.2.318) for the Stratum_technology.

Constraints:

UR1: The name shall be unique within a population of Stratum_technologies.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_technology
  SUPERTYPE OF (ONEOF (design_layer_technology,
    documentation_layer_technology))
  SUBTYPE OF (managed_design_object);
  layer_position          : layer_position_type;
  stratum_material        : ee_material;
  maximum_stratum_thickness : length_data_element;
  minimum_stratum_thickness : length_data_element;
  name                    : ee_name;
  minimum_finished_feature_size : OPTIONAL length_data_element;
  laminate_stiffness_class : OPTIONAL stiffness_class;
  minimum_finished_feature_spacing : OPTIONAL length_data_element;
  maximum_feature_size_requirement : OPTIONAL length_data_element;
  specification            : OPTIONAL process_specification;
  UNIQUE
    UR1 : name;
WHERE
  WR1: NOT(layer_position = layer_position_type.primary) OR ('AP210_ARM.DESIGN_LAYER_TECHN
    TYPEOF(SELF));
  WR2: NOT(layer_position = layer_position_type.secondary) OR ('AP210_ARM.DESIGN_LAYER_TEC
    TYPEOF(SELF));
END_ENTITY; -- stratum_technology

( *
```

4.2.872 Stratum_technology_link

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Stratum_technology_link is a type of Managed_design_object (see 4.2.524).

The data associated with a Stratum_technology_link are the following:

- precedent_stratum_technology;
- subsequent_stratum_technology;
- precedent_stratum_technology_link.

4.2.872.1 precedent_stratum_technology

The precedent_stratum_technology specifies a Stratum_technology (see 4.2.871) for the Stratum_technology_link.

4.2.872.2 subsequent_stratum_technology

The subsequent_stratum_technology specifies a Stratum_technology (see 4.2.871) for the Stratum_technology_link.

4.2.872.3 precedent_stratum_technology_link

The precedent_stratum_technology_link specifies the role of a Stratum_technology_link (see 4.2.872) for the Stratum_technology_link. The precedent_stratum_technology_link need not be specified for a particular Stratum_technology_link.

Constraints:

WR1: The precedent_stratum_technology shall not be the subsequent_stratum_technology within a population of Stratum_technology_link.

WR2: The Stratum_technology_link shall not be in a circular definition.

NOTE EXPRESS specification:

```

*)
ENTITY stratum_technology_link
--tt tc1 wd9 modified
  SUPERTYPE OF ( printed_part_template_link)
  SUBTYPE OF (managed_design_object);
    precedent_stratum_technology : stratum_technology;
    subsequent_stratum_technology : stratum_technology;
    precedent_stratum_technology_link : OPTIONAL stratum_technology_link;
WHERE
  WR1: precedent_stratum_technology :&#60;&#62;; subsequent_stratum_technology;
--tt tc1 wd9 modified
  WR2: NOT(EXISTS(precedent_stratum_technology_link)) OR
        (precedent_stratum_technology_link :&#60;&#62;; SELF);
-- Stratum_technology_link shall be acyclic.
END_ENTITY; -- stratum_technology_link

( *
```

4.2.873 Stratum_type_independent_padstack_definition

A Stratum_type_independent_padstack_definition is a type of Padstack_definition (see 4.2.595) that shall be applied to all Stratum of a design, irrespective of the type of individual Stratum.

NOTE EXPRESS specification:

```
*)
ENTITY stratum_type_independent_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- stratum_type_independent_padstack_definition
(*
```

4.2.874 Structured_text

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Structured_text is a type of Managed_design_object (see 4.2.524). A Structured_text is a portion of an Ee_document (see 4.2.316) that may have identifiable relations between itself and other portions of the Ee_document. Those relations are a container relationship in which the Structured_text is "contained in" an Ee_document, or a sequential relationship in which the Structured_text may be in a specific sequence of Structured_text within an Ee_document. Structured_text may be composed of other instances of Structured_text or Ee_text as content, in a tree structure. A Structured_text shall be acyclic. The Structured_text should at least provide content at the lowest level of the structure.

NOTE This part of ISO 10303 is exchanging the content of a document, not its presentation format.

EXAMPLE 1 An example of Structured_text is an extract from an Ee_document in SGML, in which each portion of the Ee_document has a known structure.

EXAMPLE 2 Structured_text objects can be chapters, titles, paragraphs, sentences or structured markup.

The data associated with a Structured_text are the following:

- content_markup;
- plain_text_content;
- precedent_structured_text;
- structured_text_content.

4.2.874.1 content_markup

The content_markup specifies the role of the External_definition (see 4.2.361) for the Structured_text. The content_markup identifies the computer interpretable element type of the content. Plain text shall be a string encoded according to clause 7.3.3 of ISO 10303-21. Plain text shall be identified by the following contents of the External_definition referenced by content_markup: code is "text markup tag", code_value is "plain text", reference_document is "ISO 10303-210".

EXAMPLE 1 An html encoded structure in accordance with HTML 3.2 would reference an External_definition with the following contents: code is "html tag", code_value is "body", reference_document is "!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 3.2//EN"". This example encloses the required content of a ISO 10303-21 exchange file with quotation marks ("").

4.2.874.2 plain_text_content

The plain_text_content specifies the role of the Ee_text for the Structured_text. The plain_text_content need not be provided for the Structured_text.

NOTE Plain_text_content would only not be provided if there was a need to specify lower level markup within the current markup.

4.2.874.3 precedent_structured_text

The precedent_structured_text specifies the role of the Structured_text for the Structured_text. This role is appropriate when both Structured_text are helping to compose a higher level Structured_text or are referenced by an Ee_document (see 4.2.316). The precedent_structured_text need not be provided for a Structured_text.

NOTE Precedent_structured_text is provided to ensure the correct sequencing of text contents.

4.2.874.4 structured_text_content

The structured_text_content specifies the role of the Structured_text for the Structured_text. There may be one or more entries in the structured_text_content for a particular Structured_text. The structured_text_content need not be provided for the Structured_text.

NOTE Structured_text_content would only not be provided if there was no need to specify lower level markup within the current markup or if plain_text_content were provided.

NOTE EXPRESS specification:

```
*)
ENTITY structured_text
  SUBTYPE OF (managed_design_object);
    content_markup          : external_definition;
    plain_text_content      : OPTIONAL ee_text;
    precedent_structured_text : OPTIONAL structured_text;
    structured_text_content : OPTIONAL SET [1:?] OF structured_text;
END_ENTITY; -- structured_text
```

(*

4.2.875 Styled_curve_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Styled_curve_occurrence` is a type of `Managed_design_object` (see 4.2.524). A `Styled_curve_occurrence` is an occurrence of a curve with specific parametric styling assigned.

The data associated with a `Styled_curve_occurrence` are the following:

- `of_curve_style`;
- `styled_curve`.

4.2.875.1 `of_curve_style`

The `of_curve_style` specifies the `Curve_style` (see 4.2.205) to apply to the `Curve` (see 4.2.203) for the `Styled_curve_occurrence`.

4.2.875.2 `styled_curve`

The `styled_curve` specifies the role of the `Curve` (see 4.2.203) for the `Styled_curve_occurrence`.

NOTE EXPRESS specification:

```
*)
ENTITY styled_curve_occurrence
  SUBTYPE OF (managed_design_object);
    styled_curve    : curve;
    of_curve_style  : curve_style;
END_ENTITY; -- styled_curve_occurrence

( *
```

4.2.876 `Styled_text`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Styled_text` is a type of `Shape_item` (see 4.2.818) that is formatted text that appears in the material in a `Interconnect_module` (see 4.2.440). The text shall be styled by font. The definition specifies font styling that shall be applied to realized characters. Externally defined text font and font family are supported.

NOTE This part of ISO 10303 does not standardize the display of text on computer systems.

EXAMPLE 1 Figure 105 illustrates styling that may be applied to text that appears in the material. The principal styling supported is font size through the bounding box specification. The other variations illustrated in the figure occur through modification of the text orientation during placement, and may require grouping of the text with other shape data. Some of the justification and orientation capability portrayed in the figure is provided by the `Laminate_text_string_component` (see 4.2.489)

The data associated with a `Styled_text` are the following:

— `text_font`.

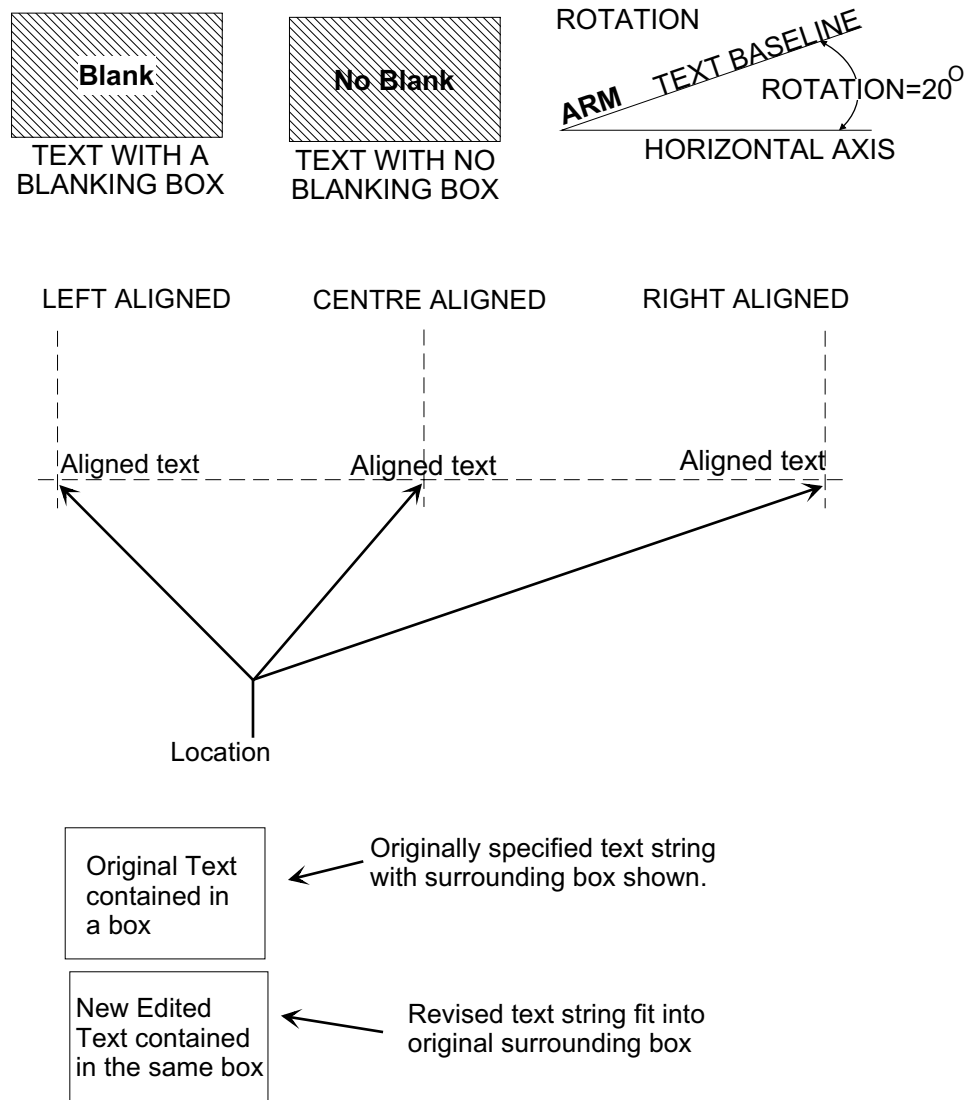


Figure 105 – `Styled_text`

4.2.876.1 `text_font`

The `text_font` specifies either the `Font_definition` (see 4.2.387) or the `Registered_font` (see 4.2.768) for the `Styled_text`.

NOTE EXPRESS specification:

```
* )
ENTITY styled_text
  SUBTYPE OF (shape_item);
  text_font : font_select;
END_ENTITY; -- styled_text

( *
```

4.2.877 Sub_assembly_relationship

A Sub_assembly_relationship is a type of Managed_design_object (see 4.2.524). A Sub_assembly_relationship is an association that specifies a subassembly constraint among two or more Assembly_component (see 4.2.34) that is aggregated into an Assembly_group_component (see 4.2.40).

NOTE 1 Industrial practice is to associate the parts included in a Assembly_group_component with members of Product_association (see 4.2.741).

NOTE 2 A group of Lands aggregated into a Footprint_occurrence do not also need to be aggregated into Padstack_occurrence.

EXAMPLE 1 Figure 106 illustrates a detail of an assembly drawing that shows components grouped into a sub-assembly, and the corresponding list of items and their associated reference_designations.

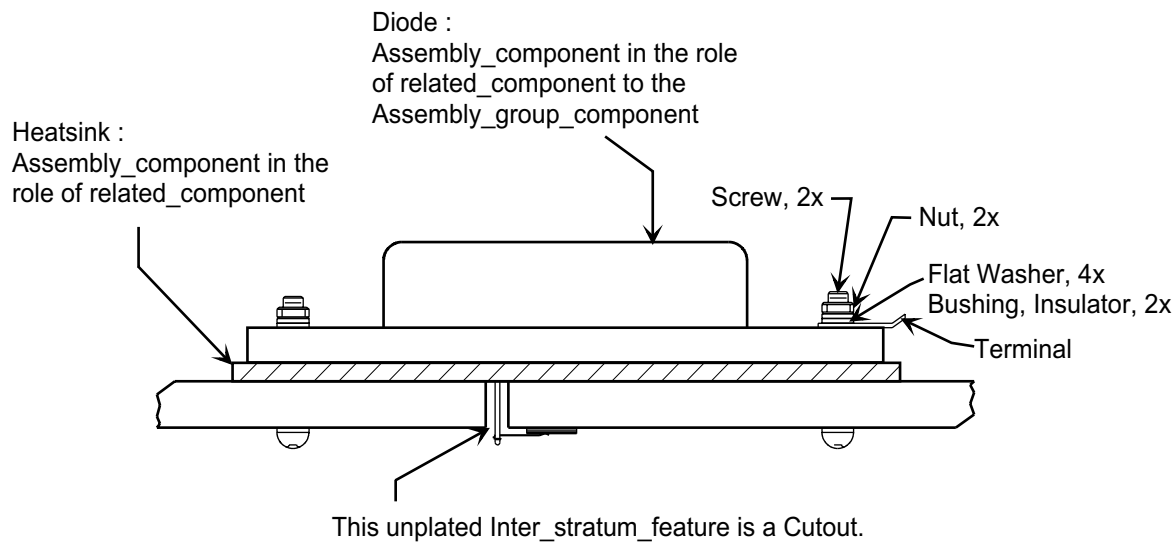
EXAMPLE 2 A Land may participate in a sub assembly where that sub assembly is a Footprint_occurrence. In that case the reference_designation provides the ability to map unambiguously to the Footprint_padstack_composition_relationship Application Object that is the reference for the location and orientation of the Land in that footprint. In that case, there is no need to populate a separate Assembly_composition_relationship between the Land and the Pcb, since the Assembly_composition_relationship between the Footprint_occurrence and the Pcb establishes the required association.

The data associated with a Sub_assembly_relationship are the following:

- sub_assembly_reference_designation;
- related_component;
- relating_component.

4.2.877.1 sub_assembly_reference_designation

The sub_assembly_reference_designation specifies the string which is the identifier for this Sub_assembly_relationship in the context of the relating_component.



Note - The mounting hardware (screw, nut, washer, bushing, terminal) individually fill the role of related_component.

A Sub_assembly_relationship relates an Assembly_component with the Assembly_group_component to which it is associated.

Items in the List	Reference Designator
Heatsink	H1
Diode	ZR1
Screw	S1
Screw	S2
Nut	N1
Nut	N2
Washer	W1
Bushing	B1
Terminal	T1
Washer	W2
Washer	W3
Washer	W4
Bushing	B2
Washer	W5

Figure 106 – Sub_assembly_relationship

4.2.877.2 related_component

The related_component specify the dependent Assembly_component (see 4.2.34) of the Sub_assembly_-relationship. There shall be two or more Assembly_component for a Sub_assembly_relationship.

4.2.877.3 relating_component

The relating_component specifies the independent Assembly_group_component (see 4.2.40) which is the aggregation of two or more Assembly_component (see 4.2.34) specified by related_component.

Constraints:

UR1: The combination of relating_component and sub_assembly_reference_designation shall be unique in a population of sub_assembly_relationship.

WR1: An Assembly_component specified as a relating_component shall not appear in the chain of components in the set of Assembly_component specified by related_component.

NOTE EXPRESS specification:

```

*)
ENTITY sub_assembly_relationship
  SUBTYPE OF (managed_design_object);
    related_component : assembly_component;
    relating_component : assembly_group_component;
    sub_assembly_reference_designation : string;
UNIQUE
  UR1: sub_assembly_reference_designation, relating_component;
END_ENTITY; -- sub_assembly_relationship
( *
```

4.2.878 Substitute_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Substitute_product is a type of Managed_design_object (see 4.2.524). A Substitute_product is an Ee_product (see 4.2.319) that may be substituted for another Ee_product within the context of an assembly or for a specific instance within an assembly. The Substitute_product is a one-way substitution only. Two instances of Substitute_product are required to support the declaration of a totally equivalent substitution relationship.

NOTE Two different substitutes may be defined within the same assembly for the same base product based on the different usages of that base product within that assembly.

The data associated with a Substitute_product are the following:

- defined_alternate_product;
- substitutable_component.

4.2.878.1 defined_alternate_product

The defined_alternate_product specifies the role of the Ee_product (see 4.2.319) for the Substitute_product. There shall be one or more Ee_product for the Substitute_product.

4.2.878.2 substitutable_component

The substitutable_component specifies the role of the Assembly_composition_relationship (see 4.2.39) for the Substitute_product.

NOTE EXPRESS specification:

```
* )
ENTITY substitute_product
  SUBTYPE OF (managed_design_object);
    defined_alternate_product : SET [1:?] OF ee_product;
    substitutable_component   : assembly_composition_relationship;
END_ENTITY; -- substitute_product

( *
```

4.2.879 Supplied_product_version

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Supplied_product_version is a type of Managed_design_object (see 4.2.524). The Supplied_product_version is a product that is obtained from an external source (the supplier).

The data associated with a Supplied_product_version are the following:

- required_certification;
- supplied_product_version;
- supplied_product_version_approval;
- supplier;
- supplier_part_number.

4.2.879.1 required_certification

The required_certification is an inverse relationship that specifies the Certification (see 4.2.94) that specifies the Supplied_product_version as its certified_product_version. There shall be no more than one Certification for a Supplied_product_version.

4.2.879.2 supplied_product_version

The supplied_product_version specifies the role of the Ee_product_version (see 4.2.328) for the Supplied_product_version.

4.2.879.3 supplied_product_version_approval

The supplied_product_version_approval specifies the role of the Ee_approval (see 4.2.314) for the Supplied_product_version.

4.2.879.4 supplier

The supplier specifies the Organization (see 4.2.569) that supplies the Supplied_product_version.

4.2.879.5 supplier_part_number

The supplier_part_number specifies the name of the Supplied_product_version. The supplier_part_number need not be provided for a particular Supplied_product_version.

Constraints:

UR1: The combination of the supplier and the supplier_part_number shall be unique within a population of Supplied_product_versions.

NOTE EXPRESS specification:

```

*)
ENTITY supplied_product_version
  SUBTYPE OF (managed_design_object);
    supplied_product_version      : ee_product_version;
    supplied_product_version_approval : ee_approval;
    supplier_part_number          : OPTIONAL ee_name;
    supplier                      : organization;
  INVERSE
    required_certification : SET [0:1] OF
                                certification FOR certified_product_version;

  UNIQUE
    UR1 : supplier, supplier_part_number;
END_ENTITY; -- supplied_product_version

( *

```

4.2.880 Surface_finish_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Surface_finish_specification is a type of Material_specification (see 4.2.537) that is a collection of the properties applicable to surface textures or coatings either for a product in-process or a completed product.

NOTE EXPRESS specification:

```

*)
ENTITY surface_finish_specification
  SUBTYPE OF (material_specification);
END_ENTITY; -- surface_finish_specification

( *

```

4.2.881 Surface_prepped_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Surface_prepped_terminal is a type of Altered_package_terminal (see 4.2.9) that defines a modification to its surface condition, for solderability or other reasons, prior to its installation on an Interconnect_module_component (see 4.2.442) during the assembly of a Pca (see 4.2.639).

The data associated with a Surface_prepped_terminal are the following:

— surface_preparation.

4.2.881.1 surface_preparation

The surface_preparation specifies the Surface_finish_specification (see 4.2.880) for modification to the surface finish of the Surface_prepped_terminal.

NOTE EXPRESS specification:

```
*)
ENTITY surface_prepped_terminal
  SUBTYPE OF (altered_package_terminal);
    surface_preparation : surface_finish_specification;
END_ENTITY; -- surface_prepped_terminal
```

(*

4.2.882 Symmetrical_tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Symmetrical_tolerance_characteristic is a type of Tolerance_characteristic (see 4.2.911) that has symmetrical tolerances around the nominal value.

The data associated with a Symmetrical_tolerance_characteristic are the following:

— component_characteristic.

4.2.882.1 component_characteristic

The component_characteristic specifies the value of the Numeric_parameter (see 4.2.565) for the Symmetrical_tolerance_characteristic. There shall be exactly two Numeric_parameter for the Symmetrical_tolerance_characteristic.

NOTE EXPRESS specification:

```
*)
ENTITY symmetrical_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET [2:2] OF
                                                                numeric_parameter;
END_ENTITY; -- symmetrical_tolerance_characteristic
```

(*

4.2.883 Symmetry_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Symmetry_tolerance is a type of Geometric_tolerance_with_referenced_datum_system (see 4.2.422) that specifies or partially specifies the design requirement on the symmetry of a Shape_element (see 4.2.810) with respect to a Datum_plane (see 4.2.232). That is, the median points of all opposed or correspondingly located elements of two or more surfaces of a Shape_element must fall within the defined Tolerance_zone_boundary (see 4.2.915). The Tolerance_zone_boundary is an opposing plane set that is symmetrically disposed about the specified Datum_plane.

NOTE Symmetry tolerance is defined in terms of tolerance zone definitions in 17.12 of ISO/DIS 1101.

Constraints:

WR1: The Shape_element specified as the toleranced_shape_element shall be a Centre_plane.

NOTE The attribute toleranced_shape_element is declared in the Geometric_tolerance entity of which this entity is a subtype.

WR2: There shall be at most two Datum in the Datum_system the definition of which is specified by the datum_system_definition.

NOTE The attribute datum_system_definition is declared in the Geometric_tolerance_with_referenced_datum_system entity of which this entity is a subtype.

WR3: There shall be a Datum_plane in the Datum_system the definition of which is specified by the datum_system_definition.

WR4: The Symmetry_tolerance shall specify a Tolerance_zone_definition as its tolerance_zone_or_boundary_definition; and that Tolerance_zone_definition shall specify a Tolerance_zone_boundary_definition_with_specified_size as its boundary_definition; and that Tolerance_zone_boundary_definition_with_specified_size shall specify an Opposing_boundary_set_size_characteristic as its specified_boundary_size.

IP1: The form of the toleranced Shape_element shall be consistent with the form of a shape element for which the application of a symmetry tolerance is sensible.

NOTE EXPRESS specification:

```
* )
ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- symmetry_tolerance
```

(*

4.2.884 Tangent_plane

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tangent_plane` is a type of `Tangent_shape_element` (see 4.2.885) that is planar in form.

EXAMPLE 1 An `Angularity_tolerance` (see 4.2.26) might be applied to the `Tangent_plane` derived from a surface of a `Physical_unit` (see 4.2.664), as opposed to the surface itself, if variations in the form of the surface are unimportant or are sufficiently constrained by other tolerances.

Constraints:

IP1: The form of the `Shape_element` from which the `Tangent_plane` is derived shall be consistent with the form of a shape element from which a tangent plane can be derived.

NOTE EXPRESS specification:

```
*)
ENTITY tangent_plane
  SUBTYPE OF (tangent_shape_element);
END_ENTITY; -- tangent_plane

( *
```

4.2.885 `Tangent_shape_element`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tangent_shape_element` is a type of `Derived_shape_element` (see 4.2.260) that is either a straight line that contacts a curve at only the high points or is a plane that contacts a surface at only its high points. The `Shape_element` from which the `Tangent_shape_element` is derived is associated with the `Tangent_shape_element` by the `Shape_element_deriving_relationship` (see 4.2.814) that is specified as the `deriving_relationships`. A `Tangent_shape_element` may be a `Tangent_plane` (see 4.2.884).

NOTE The attribute `deriving_relationships` is declared in the `Derived_shape_element` entity of which this entity is a subtype.

Constraints:

WR1: The `Tangent_shape_element` shall be derived from only one `Shape_element`.

IP1: The `Shape_element` from which the `Tangent_shape_element` is derived shall be a curve or surface.

NOTE EXPRESS specification:

```
* )  
ENTITY tangent_shape_element  
  SUPERTYPE OF (tangent_plane)  
  SUBTYPE OF (derived_shape_element);  
END_ENTITY; -- tangent_shape_element  
  
(*
```

4.2.886 Technology

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Technology is a type of Managed_design_object (see 4.2.524). A Technology is the manufacturing process for the realization of an Ee_product (see 4.2.319).

The data associated with a Technology are the following:

- classification;
- material_product_system;
- process_technology.

4.2.886.1 classification

The classification specifies the External_definition (see 4.2.361) which defines the Technology.

4.2.886.2 material_product_system

The material_product_system specifies the Ee_material (see 4.2.318) which defines the material combination which is necessary for the output of the manufacturing process.

4.2.886.3 process_technology

The process_technology specifies the Process_specification (see 4.2.740) which defines the exact process of the Technology.

NOTE EXPRESS specification:

```

*)
ENTITY technology
  SUBTYPE OF (managed_design_object);
    process_technology      : process_specification;
    classification          : external_definition;
    material_product_system : ee_material;
END_ENTITY; -- technology

--tt tc1 correction
--mtk corrected WR1:
( *
```

4.2.887 Template_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Template_definition is a type of Ee_product_definition (see 4.2.323).

The data associated with a Template_definition are the following:

— associated_template_occurrence.

4.2.887.1 associated_template_occurrence

The associated_template_occurrence is an inverse relationship that specifies that the existence of the Template_definition is dependent on the existence of a Part_template to reference the Template_definition.

NOTE EXPRESS specification:

```

*)
ENTITY template_definition
  SUBTYPE OF (ee_product_definition);
INVERSE
  associated_template_occurrence :
    part_template FOR associated_definition;
END_ENTITY; -- template_definition
( *
```

4.2.888 Template_material_cross_section_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Template_material_cross_section_boundary is a type of Non_feature_shape_element (see 4.2.560).

The data associated with a `Template_material_cross_section_boundary` are the following:

— `associated_definition`.

4.2.888.1 `associated_definition`

The `associated_definition` specifies a `Printed_part_template_material` (see 4.2.735) for the `Template_material_cross_section_boundary`.

Constraints:

UR1: The combination of `associated_definition` and the name attribute inherited from `Shape_element` (see 4.2.810) shall be unique within a population of `Template_material_cross_section_boundary`.

WR1: The name attribute inherited from `Shape_element` (see 4.2.810) shall be one of "top", "bottom", "left", or "right" within a population of `Template_material_cross_section_boundary`.

NOTE EXPRESS specification:

```
*)
ENTITY template_material_cross_section_boundary
  SUBTYPE OF (non_feature_shape_element);
  associated_definition : printed_part_template_material;
UNIQUE
  UR1: associated_definition, SELF\shape_element.name;
WHERE
  WR1: SELF\shape_element.name IN ['top', 'bottom', 'left', 'right'];
END_ENTITY; -- template_material_cross_section_boundary

(*
```

4.2.889 `Template_stratum_assignment`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Template_stratum_assignment` is a type of `Managed_design_object` (see 4.2.524). A `Template_stratum_assignment` is the relationship between a `Stratum` (see 4.2.845) and any one of an `Stratum_feature_template` (see 4.2.861), a `Material_addition_feature_template` (see 4.2.529), or a `Material_removal_feature_template` (see 4.2.534) for the purpose of binding a template to a specific `Stratum` for a `Padstack_definition`.

The data associated with a `Template_stratum_assignment` are the following:

— `bound_template`;

— `bound_stratum`.

4.2.889.1 bound_template

The bound_template specifies any one of the Stratum_feature_template (see 4.2.861), Material_addition_feature_template (see 4.2.529), Material_removal_feature_template (see 4.2.534) as the assigned object for the Template_stratum_assignment.

4.2.889.2 bound_stratum

The bound_stratum specifies the role of the Stratum (see 4.2.845) for the Template_stratum_assignment.

NOTE EXPRESS specification:

```
*)
ENTITY template_stratum_assignment
  SUBTYPE OF (managed_design_object);
    bound_template : template_select;
    bound_stratum  : stratum;
END_ENTITY; -- template_stratum_assignment

(*
```

4.2.890 Terminal_pair_port_equivalence_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Terminal_pair_port_equivalence_relationship is a type of Managed_design_object (see 4.2.524). A Terminal_pair_port_equivalence_relationship is the assignment association between a pair of scalar ports and a single distributed port.

The data associated with a Terminal_pair_port_equivalence_relationship are the following:

- equivalent_port;
- scalar_ports.

4.2.890.1 equivalent_port

The equivalent_port specifies the Analytical_model_port (see 4.2.17) that represents the single distributed port for the Terminal_pair_port_equivalence_relationship.

4.2.890.2 scalar_ports

The scalar_ports specify the two Analytical_model_scalar_port (see 4.2.20) in the ordered sequence for the Terminal_pair_port_equivalence_relationship. The second scalar_port is always the reference port. Other details depend on the specific domain of the Analytical_model (see 4.2.14).

EXAMPLE 1 Port "INPUT" of a microwave coaxial filter model is the equivalent_port for this example.

Port "INPUT center pin" is the first Analytical_model_scalar_port for this example. Port "INPUT shield" is the second Analytical_model_scalar_port for this example.

Constraints:

WR1: The distributed_properties attribute of the Analytical_model_port in the role of equivalent_port shall be "TRUE".

WR2: The scalar_ports and equivalent_port shall not reference the same Analytical_model.

WR3: The scalar_ports shall reference the same Analytical_model.

WR4: The equivalent_port shall not be a member of Digital_analytical_model_port.

NOTE EXPRESS specification:

```

*)
ENTITY terminal_pair_port_equivalence_relationship
  SUBTYPE OF (managed_design_object);
    equivalent_port : analytical_model_port;
    scalar_ports    : LIST [2:2] OF UNIQUE analytical_model_scalar_port;
END_ENTITY; -- terminal_pair_port_equivalence_relationship

( *
```

4.2.891 Terminal_swap_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Terminal_swap_specification is a type of Change_design_object_management_relationship (see 4.2.95). The Terminal_swap_specification is part of the swapping specification from the previous design definition to the current design definition. It identifies the terminal in the previous design that was connected to a node, and the terminal in the current design that is connected to that node instead.

NOTE With the pin mapping between the functional and physical representations in the library, this is sufficient information to derive the swapping at the component terminal; however, note that this is in the previous and present design definitions and not in the library functional definition.

The data associated with a Terminal_swap_specification are the following:

- current_design_object;
- previous_design_object.

4.2.891.1 current_design_object

The current_design_object specifies the Functional_unit_terminal_node_assignment (see 4.2.409) of the terminal node after the swap for the Terminal_swap_specification.

4.2.891.2 previous_design_object

The previous_design_object specifies the Functional_unit_terminal_node_assignment (see 4.2.409) of the terminal node before the swap for the Terminal_swap_specification.

Constraints:

WR1: The current_design_object shall not be the previous_design_object.

NOTE EXPRESS specification:

```
*)
ENTITY terminal_swap_specification
  SUBTYPE OF (change_design_object_management_relationship);
    SELF\change_design_object_management_relationship.previous_design_object :
functional_unit_terminal_node_assignment;
    SELF\change_design_object_management_relationship.current_design_object :
functional_unit_terminal_node_assignment;
END_ENTITY; -- terminal_swap_specification
```

(*

4.2.892 Termination_constraint

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Termination_constraint is a type of Managed_design_object (see 4.2.524). A Termination_constraint is a constraint that requires a set of two or more Mating_connector_termination (see 4.2.540) to satisfy an Ee_requirement_occurrence (see 4.2.330).

The data associated with a Termination_constraint are the following:

- constrained_termination;
- termination_usage_constraint.

4.2.892.1 constrained_termination

The constrained_termination specifies the role of the Mating_connector_termination (see 4.2.540) for the Termination_constraint. There shall be two or more Mating_connector_termination for the Termination_constraint.

4.2.892.2 termination_usage_constraint

The termination_usage_constraint specifies the role of the Ee_requirement_occurrence (see 4.2.330) for the Termination_constraint.

NOTE EXPRESS specification:

```
*)
ENTITY termination_constraint
  SUBTYPE OF (managed_design_object);
    constrained_termination      : SET [2:?] OF mating_connector_termination;
    termination_usage_constraint : ee_requirement_occurrence;
END_ENTITY; -- termination_constraint

(*
```

4.2.893 Tertiary_orientation_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tertiary_orientation_feature is both a Part_feature (see 4.2.609) and a Datum_feature (see 4.2.226). A tertiary_orientation_feature provides anti-rotation capability for unambiguous orientation of the part in the datum_reference_frame.

The data associated with a Tertiary_orientation_feature are the following:

— associated_body_vertical_extent.

4.2.893.1 associated_body_vertical_extent

An associated_body_vertical_extent specifies the role of the Package_body_surface (see 4.2.578) for a Tertiary_orientation_feature. The associated_body_vertical_extent helps to identify the surface or surfaces which contain the feature. An associated_body_vertical_extent need not be specified for a Tertiary_orientation_feature.

NOTE EXPRESS specification:

```
*)
ENTITY tertiary_orientation_feature
  SUBTYPE OF (part_feature, datum_feature);
    associated_body_vertical_extent : OPTIONAL SET [1:2] OF package_body_surface;
END_ENTITY; -- tertiary_orientation_feature

( *
```

4.2.894 Test_point_part_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Test_point_part_feature is a type of Part_tooling_feature (see 4.2.633). A Test_point_part_feature identifies a feature for use with test instrumentation.

NOTE A Test_point_part_feature may be related to a Probe_access_area (see 4.2.739) if the layout view is also supplied.

NOTE EXPRESS specification:

```
*)
ENTITY test_point_part_feature
  SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- test_point_part_feature

( *
```

4.2.895 Test_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Test_requirement_allocation is a type of Requirement_allocation (see 4.2.769) for which the Ee_requirement_occurrence (see 4.2.330) has test characteristics. A Test_requirement_allocation is the relationship between the Test_requirement_occurrence (see 4.2.896) and the Ee_product_version (see 4.2.328) to which the requirement is allocated. The terminals of the product are related to either parametric data in an executable form or to data derived from experimental or theoretical measurements.

NOTE The structural split of allocation and requirement occurrence permits re-use of common test requirements and associated procedures over multiple products.

The data associated with a Test_requirement_allocation are the following:

- target_product_object;
- analytical_model_based_data;
- functional_specification_based_data;
- allocated_requirement.

4.2.895.1 target_product_object

The target_product_object specifies the role of the Test_component (see ??) for the Test_requirement_allocation. The target_product_object is a specialization of the attribute inherited from Assembly_requirement_allocation.

4.2.895.2 analytical_model_based_data

The analytical_model_based_data specifies the role of the Part_feature_model_port_assignment (see 4.2.611) for the Test_requirement_allocation. The analytical_model_based_data specifies the mapping of the ports of the Analytical_model (see 4.2.14) onto the relevant terminals of the product for which the test is specified.

NOTE The mapping provides a formal mechanism for associating the details of the required behavior represented in computer executable form, as defined in the body of an Analytical_model, with the terminals of the tested product.

4.2.895.3 functional_specification_based_data

The functional_specification_based_data specifies the role of the Functional_unit_usage_view_to_part-terminal_assignment (see ??) for the Test_requirement_allocation. The functional_specification_based_data specifies the mapping of the terminals of the Functional_unit_usage_view (see 4.2.410) onto the relevant terminals of the product for which the test is specified.

NOTE The mapping provides a formal mechanism for associating the details of the required behavior represented as a result of an experiment, as defined by a Functional_specification, with the terminals of the tested product.

4.2.895.4 allocated_requirement

The allocated_requirement specifies the role of the Test_requirement_occurrence (see 4.2.896) for the Test_requirement_allocation. The allocated_requirement is a specialization of the attribute inherited from Assembly_requirement_allocation.

NOTE EXPRESS specification:

```

*)
ENTITY test_requirement_allocation
  SUBTYPE OF (requirement_allocation);
    SELF\requirement_allocation.target_product_object : ee_product_version;
    SELF\requirement_allocation.allocated_requirement :
test_requirement_occurrence;
    analytical_model_based_test_access : OPTIONAL SET [1:?] OF part_feature_model_port_ass.
    functional_specification_based_test_access : OPTIONAL SET [1:?] OF functional_usage_vi
WHERE
  WR1: NOT(EXISTS(analytical_model_based_test_access)) OR
    (SIZEOF(QUERY(ta &#60;* analytical_model_based_test_access |
    NOT(ta.source.associated_definition.version =
      SELF\requirement_allocation.target_product_object)
    )) = 0);
  WR2: NOT(EXISTS(analytical_model_based_test_access)) OR
    (SIZEOF(QUERY(ta &#60;* analytical_model_based_test_access |
    NOT(ta.target.accessed_analytical_model =
      SELF\requirement_allocation.allocated_requirement.required_analytical_representation
    )) = 0);
  WR3: EXISTS(analytical_model_based_test_access) XOR EXISTS(functional_specification_based
  WR4: NOT(EXISTS(analytical_model_based_test_access)) OR
    (EXISTS(SELF\requirement_allocation.allocated_requirement\ee_requirement_occurrence
  WR5: NOT(EXISTS(functional_specification_based_test_access)) OR
    (SIZEOF(QUERY(fa &#60;* functional_specification_based_test_access |
    NOT(fa.physical_usage_view_terminal\part_feature.associated_definition.version =
      SELF\requirement_allocation.target_product_object)
    )) = 0);
  WR6: NOT(EXISTS(functional_specification_based_test_access)) OR
    (EXISTS(SELF\requirement_allocation.
      allocated_requirement\ee_requirement_occurrence.
      required_functional_specification));
  WR7: NOT(EXISTS(functional_specification_based_test_access)) OR
    (SIZEOF(QUERY(fa &#60;* functional_specification_based_test_access |
    NOT(fa.functional_usage_view_terminal\functional_unit_usage_view_terminal_definition.
      associated_functional_unit_definition =
      SELF\requirement_allocation.
      allocated_requirement\ee_requirement_occurrence.
      required_functional_specification.characterizing_functional_unit.definition)
    )) = 0);
END_ENTITY; -- test_requirement_allocation
(

```

4.2.896 Test_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Test_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that specifies a test related requirement. The test method shall be specified in the requirement_specification inherited from Ee_requirement_occurrence. The Test_requirement_occurrence provides a representation of minimum bounds on behaviour. In order to relate a Test_requirement_occurrence to a product, a Requirement_allocation is required to be exchanged that references this Application Object and the version of the product to which the test requirement is allocated. That Requirement_allocation shall also specify a number of Part_feature_analytical_model_port_assignment in order to specify the measurement access points for the test. It is valid to exchange a Test_requirement_occurrence without a Requirement_allocation Either an Analytical_representation shall be provided or a Functional_specification shall be provided.

NOTE 1

NOTE 2 This edition of this part of ISO 10303 does not explicitly relate the test requirement to the test setup.

NOTE 3 The Test_requirement_occurrence may be used as a reference for Planned_characteristic.

NOTE 4 The Test_requirement_occurrence may be a derived requirement from a Design_requirement_occurrence, in which case production capability based margins would cause the actual characteristics to be different between the Design_requirement_occurrence and the Test_requirement_occurrence.

NOTE EXPRESS specification:

```
*)
ENTITY test_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
WHERE
  WR1: SIZEOF(SELF\ee_requirement_occurrence.required_analytical_representation) +
        SIZEOF(SELF\ee_requirement_occurrence.required_functional_specification) > 0;
END_ENTITY; -- test_requirement_occurrence
(*
```

4.2.897 Test_select_product

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Test_select_product is a type of Alternate_select_product (see 4.2.12) in which the selection among alternate choices is performed as a result of a test operation, in order to optimize the performance of the Ee_product (see 4.2.319).

NOTE EXPRESS specification:

```
*)
ENTITY test_select_product
  SUBTYPE OF (alternate_select_product);
END_ENTITY; -- test_select_product

( *
```

4.2.898 Test_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Test_specification is a type of Ee_specification (see 4.2.338) that provides information about the physical or functional testing of an Ee_product (see 4.2.319).

NOTE EXPRESS specification:

```
*)
ENTITY test_specification
  SUBTYPE OF (ee_specification);
END_ENTITY; -- test_specification

( *
```

4.2.899 Theoretically_exact_dimension_representation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Theoretically_exact_dimension_representation is a type of Singular_dimension_representation (see 4.2.829) that describes the theoretically exact size, form, orientation, or location of one or more Shape_element (see 4.2.810).

NOTE 1 Theoretically exact dimensions are described in clause 10 of ISO 1101.

NOTE 2 The term in ASME Y14.5M corresponding to Theoretically_exact_dimension_representation is basic dimension. Basic dimensions are defined in 1.3.9 of ASME Y14.5M.

NOTE EXPRESS specification:

```
*)
ENTITY theoretically_exact_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- theoretically_exact_dimension_representation

( *
```

4.2.900 Thermal_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_component is a type of Assembly_component (see 4.2.34) that is included in the design in order to satisfy a heat flow requirement.

NOTE An Assembly_component utilized as a Thermal_component may be another type of Assembly_component as well.

EXAMPLE 1 Examples of Thermal_component are: a heatsink, a thermal via, an active thermal cooler (such as a fan), or a thermoelectric cooler.

NOTE EXPRESS specification:

```
*)
ENTITY thermal_component
  SUBTYPE OF (assembly_component);
END_ENTITY; -- thermal_component

( *
```

4.2.901 Thermal_feature

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_feature is a type of Part_feature (see 4.2.609) that is specified as having a role in the exchange of heat with the application environment of the Physical_unit (see 4.2.664).

NOTE EXPRESS specification:

```
*)
ENTITY thermal_feature
  SUBTYPE OF (part_feature);
END_ENTITY; -- thermal_feature

( *
```

4.2.902 Thermal_isolation_laminate_component

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Thermal_isolation_laminate_component` is a type of `Design_intent_modification` (see 4.2.267) that indicates that material is to be removed in the realization of the feature that is in the role of `design_intent` inherited from `Design_intent_modification`. A `Thermal_isolation_laminate_component` may be a `Filled_area_material_removal_structure` (see 4.2.385). The requirement shall be interpreted mathematically as a requirement for application of the regularized Boolean subtraction operation.

The data associated with a `Thermal_isolation_laminate_component` are the following:

— `usage_definition`.

4.2.902.1 `usage_definition`

The `usage_definition` specifies the role of the `Thermal_isolation_removal_template` (see 4.2.904) for the `Thermal_isolation_laminate_component`.

NOTE EXPRESS specification:

```
*)
ENTITY thermal_isolation_laminate_component
  SUBTYPE OF (filled_area_material_removal_laminate_component);
  SELF\assembly_component.usage_definition : thermal_isolation_removal_template;
END_ENTITY; -- thermal_isolation_laminate_component

(*
```

4.2.903 `Thermal_isolation_removal_structure`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Thermal_isolation_removal_structure` is a type of `Filled_area_material_removal_structure` (see 4.2.385) included in the design for reasons of thermal isolation.

EXAMPLE 1 Figure 107 illustrates the removal of material from a power plane based on the shape of this structure.

The data associated with a `Thermal_isolation_removal_structure` are the following:

— `thermal_isolation_spacing_requirement`.

4.2.903.1 `thermal_isolation_spacing_requirement`

The `thermal_isolation_spacing_requirement` is an inverse relationship that specifies that the existence of the `Thermal_isolation_removal_structure` is dependent on the existence of the `Thermal_isolation_requirement_assignment` (see 4.2.905) that specifies the `Thermal_isolation_removal_structure` as its resulting structure.

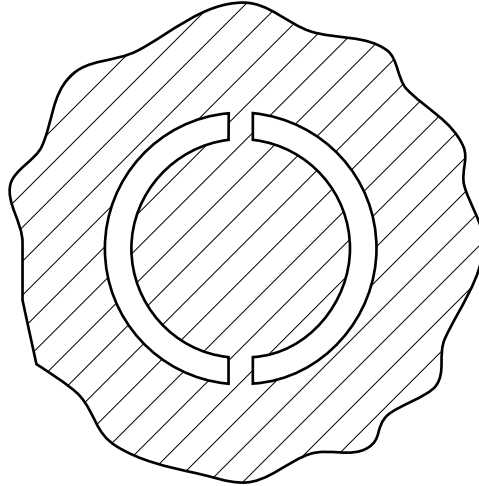


Figure 107 – Thermal_isolation_removal_structure

NOTE EXPRESS specification:

```

*)
ENTITY thermal_isolation_removal_structure
  SUBTYPE OF (filled_area_material_removal_structure);
  INVERSE

    thermal_isolation_spacing_requirement :
    thermal_isolation_requirement_assignment FOR resulting_structure;
END_ENTITY; -- thermal_isolation_removal_structure

( *
```

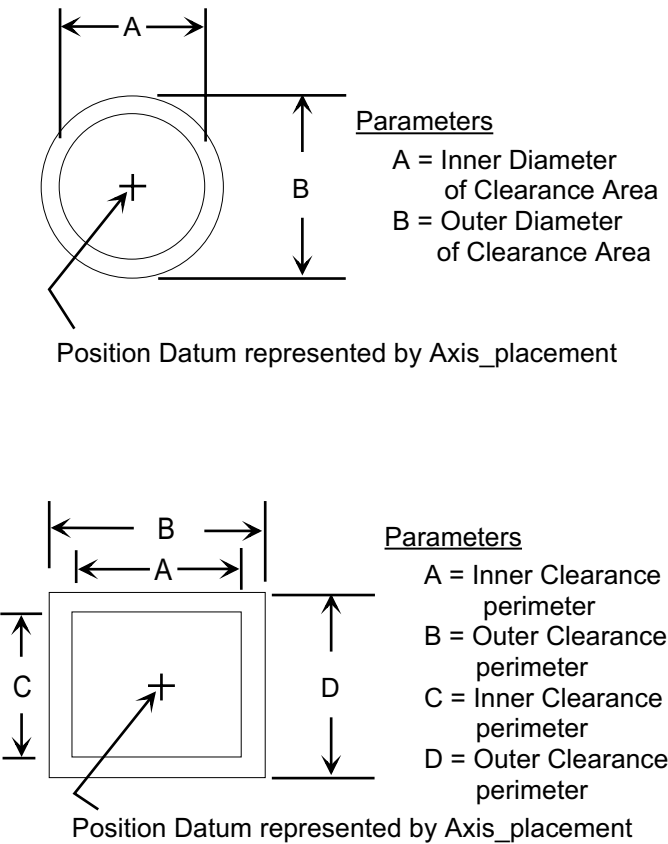
4.2.904 Thermal_isolation_removal_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_isolation_removal_template is a type of Part_template (see 4.2.620) that describes the shape of material to be removed from a feature. The purposes predefined in this part of ISO 10303 include electrical and thermal isolation.

EXAMPLE 1 Figure 108 illustrates a template and associated parameters designed for electrical isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values.

EXAMPLE 2 Figure 109 illustrates a template and associated parameters designed for thermal isolation. For a specific template, values will be assigned for the parameters and the explicit geometry will be representative of the assigned parameter values. In this example, an instance would be a complex instance including Composite_unit_shape_element (see 4.2.167) and Thermal_isolation_removal_template.

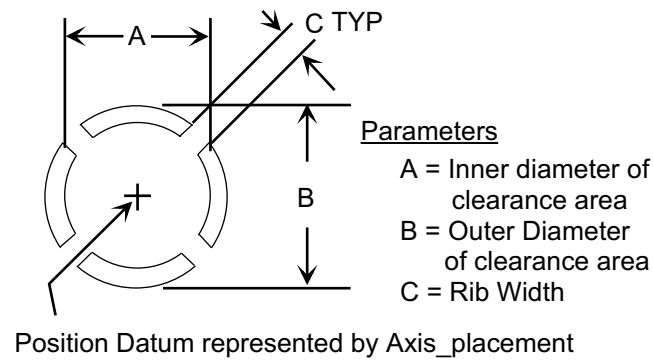
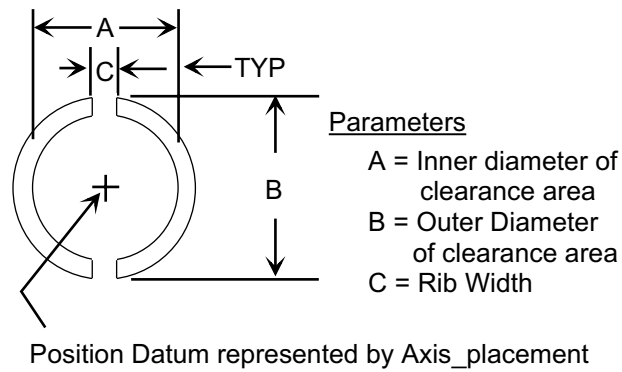


Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Electrical Isolation

Figure 108 – Thermal_isolation_removal_template supports electrical isolation



Physical characteristics are inherited from Part_template.

Reference_document is : IEC 61182-1

Note : Thermal isolation

Figure 109 – Thermal_isolation_removal_template supports thermal isolation and electrical conduction

NOTE EXPRESS specification:

```

*)
ENTITY thermal_isolation_removal_template
  SUBTYPE OF (material_removal_feature_template);
INVERSE
  thermal_isolation_spacing_requirement : thermal_isolation_requirement_assignment FOR res
END_ENTITY; -- thermal_isolation_removal_template

( *
```

4.2.905 Thermal_isolation_requirement_assignment

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_isolation_requirement_assignment is a type of Managed_design_object (see 4.2.524). A Thermal_isolation_requirement_assignment is the relationship among requirements for thermal isolation and the resulting structure of the Interconnect_module (see 4.2.440) or the template.

The data associated with a Thermal_isolation_requirement_assignment are the following:

- angular_orientation_requirement;
- effective_current_capacity_requirement;
- resulting_structure;
- thermal_isolation_spacing_requirement.

4.2.905.1 angular_orientation_requirement

The angular_orientation_requirement specifies one role of the Ee_requirement_occurrence (see 4.2.330) for the Thermal_isolation_requirement_assignment.

4.2.905.2 effective_current_capacity_requirement

The effective_current_capacity_requirement specifies one role of the Ee_requirement_occurrence (see 4.2.330) for the Thermal_isolation_requirement_assignment.

4.2.905.3 resulting_structure

The resulting_structure specifies the role of the Thermal_isolation_removal_structure (see 4.2.903) or of the Thermal_isolation_removal_template (see 4.2.904) for the Thermal_isolation_requirement_assignment.

4.2.905.4 thermal_isolation_spacing_requirement

The thermal_isolation_spacing_requirement specifies one role of the Ee_requirement_occurrence (see 4.2.330) for the Thermal_isolation_requirement_assignment.

NOTE EXPRESS specification:

```

*)
ENTITY thermal_isolation_requirement_assignment
  SUBTYPE OF (managed_design_object);
    resulting_structure :
      thermal_isolation_removal_structure_or_template;
    thermal_isolation_spacing_requirement : layout_spacing_requirement_occurrence;
    effective_current_capacity_requirement : ee_requirement_occurrence;
    angular_orientation_requirement      : ee_requirement_occurrence;
END_ENTITY; -- thermal_isolation_requirement_assignment

( *
```

4.2.906 Thermal_network_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_network_definition is a type of Functional_unit_network_definition (see 4.2.402) for which the network nodal formulation is a representation of heat flow rather than current flow.

NOTE EXPRESS specification:

```

*)
ENTITY thermal_network_definition
  SUBTYPE OF (functional_unit_network_definition);
END_ENTITY; -- thermal_network_definition

( *
An approved thermal_network_definition shall be consistent with conservative laws.
( *
```

4.2.907 Thermal_requirement_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_requirement_allocation is a type of Assembly_requirement_allocation (see 4.2.53) for which the Ee_requirement_occurrence (see 4.2.330) has thermal characteristics. A Thermal_requirement_allocation is the relationship between the Thermal_requirement_occurrence (see 4.2.908) and the Thermal_component (see 4.2.900) that satisfies it.

The data associated with a Thermal_requirement_allocation are the following:

- assembly_assignment;
- requirement.

4.2.907.1 assembly_assignment

The assembly_assignment specifies the role of the Thermal_component (see 4.2.900) for the Thermal_requirement_allocation.

4.2.907.2 requirement

The requirement specifies the role of the Thermal_requirement_occurrence (see 4.2.908) for the Thermal_requirement_allocation.

NOTE EXPRESS specification:

```

*)
ENTITY thermal_requirement_allocation
  SUBTYPE OF (assembly_requirement_allocation);
    SELF\assembly_requirement_allocation.assembly_assignment : thermal_component;
    SELF\assembly_requirement_allocation.requirement          :
thermal_requirement_occurrence;
END_ENTITY; -- thermal_requirement_allocation

( *
```

4.2.908 Thermal_requirement_occurrence

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermal_requirement_occurrence is a type of Ee_requirement_occurrence (see 4.2.330) that specifies a heat transfer related requirement.

NOTE EXPRESS specification:

```

*)
ENTITY thermal_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- thermal_requirement_occurrence

( *
```

4.2.909 Thermodynamic_temperature_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Thermodynamic_temperature_data_element is a

The data associated with a Thermodynamic_temperature_data_element are the following:

— additional_data.

4.2.909.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)  
ENTITY thermodynamic_temperature_data_element  
    SUBTYPE OF (data_element);  
END_ENTITY; -- thermodynamic_temperature_data_element  
  
(*
```

4.2.910 Time_data_element

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Time_data_element is a

The data associated with a Time_data_element are the following:

— additional_data.

4.2.910.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY time_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- time_data_element

( *
```

4.2.911 Tolerance_characteristic

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_characteristic is a type of Coordinated_characteristic (see 4.2.197). A Tolerance_characteristic may be one of Maximum_tolerance_characteristic (see 4.2.542), Minimum_tolerance_characteristic (see 4.2.548), Plus_minus_tolerance_characteristic (see 4.2.691), Statistical_tolerance_characteristic (see 4.2.842), or Symmetrical_tolerance_characteristic (see 4.2.882).

The data associated with a Tolerance_characteristic are the following:

— component_characteristic.

4.2.911.1 component_characteristic

The component_characteristic specifies the role of the Numeric_parameter (see 4.2.565) for the Tolerance_characteristic. There shall be one or more Numeric_parameter for the Tolerance_characteristic.

NOTE Members of this application object are not exchanged to support geometric tolerancing.

NOTE EXPRESS specification:

```
*)
ENTITY tolerance_characteristic
  SUPERTYPE OF (ONEOF (minimum_tolerance_characteristic,
    statistical_tolerance_characteristic,
    symmetrical_tolerance_characteristic,
    maximum_tolerance_characteristic,
    plus_minus_tolerance_characteristic))
  SUBTYPE OF (coordinated_characteristic);
  SELF\coordinated_characteristic.component_characteristic : SET [1:?] OF
                                                                numeric_parameter;
END_ENTITY; -- tolerance_characteristic

( *
```

4.2.912 Tolerance_specific_restraint_condition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

The `Tolerance_specific_restraint_condition` is a type of `Restraint_condition` that is only applied to specified instances of `Geometric_tolerance` (see 4.2.419) and `Dimensional_characteristic` (see 4.2.298).

The data associated with a `Tolerance_specific_restraint_condition` are the following:

— `affected_tolerances`.

4.2.912.1 `affected_tolerances`

The `affected_tolerances` specifies the `Geometric_tolerance` (see 4.2.419) and `Dimensional_characteristic` (see 4.2.298) to which the restraint condition applies. There shall be one or more `Geometric_tolerance` or `Dimensional_characteristic` specified as the `affected_tolerances` by a `Tolerance_specific_restraint_condition`.

NOTE EXPRESS specification:

```
* )
ENTITY tolerance_specific_restraint_condition
  SUBTYPE OF (restraint_condition);
  affected_tolerances : SET [1:?] OF geometric_or_dimension_select;
END_ENTITY; -- tolerance_specific_restraint_condition

( *
```

4.2.913 `Tolerance_zone`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tolerance_zone` is a type of `Non_feature_shape_element` (see 4.2.560) that is a region within which the toleranced `Shape_element` (see 4.2.810) must lie.

NOTE Tolerance zones are defined in clause 7 of ISO 1101.

The data associated with a `Tolerance_zone` are the following:

— `zone_boundary`;

— `zone_definition`.

4.2.913.1 `zone_boundary`

The `zone_boundary` specifies the `Tolerance_zone_boundary` (see 4.2.915) that define the limits of the `Tolerance_zone`.

4.2.913.2 zone_definition

The zone_definition is an inverse relationship that specifies that the existence of the Tolerance_zone is dependent on the existence of the Tolerance_zone_definition (see 4.2.920) that specifies the Tolerance_zone as its defined_tolerance_zone.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone
  SUBTYPE OF (non_feature_shape_element);
  zone_boundary : tolerance_zone_boundary;
  INVERSE
    zone_definition : tolerance_zone_definition FOR defined_tolerance_zone;
END_ENTITY; -- tolerance_zone

( *
```

4.2.914 Tolerance_zone_and_viewing_plane_orienting_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_and_viewing_plane_orienting_relationship is a type of Managed_design_object (see 4.2.524). A Tolerance_zone_and_viewing_plane_orienting_relationship is a relationship that associates a Tolerance_zone (see 4.2.913) with the Viewing_plane (see 4.2.956) by which it is oriented.

The data associated with a Tolerance_zone_and_viewing_plane_orienting_relationship are the following:

- oriented_tolerance_zone;
- orienting_viewing_plane;
- zone_orientation.

4.2.914.1 oriented_tolerance_zone

The oriented_tolerance_zone specifies the Tolerance_zone (see 4.2.913) that is oriented with respect to the Viewing_plane (see 4.2.956) specified as the orienting_viewing_plane.

4.2.914.2 orienting_viewing_plane

The orienting_viewing_plane specifies the Viewing_plane (see 4.2.956) by which the Tolerance_zone (see 4.2.913) that is specified as the oriented_tolerance_zone is oriented.

4.2.914.3 zone_orientation

The zone_orientation is an inverse relationship that specifies that the existence of the Tolerance_zone_and_viewing_plane_orienting_relationship is dependent on the existence of the Viewing_plane_based_orientation (see 4.2.957) that specifies the Tolerance_zone_and_viewing_plane_orienting_relationship as its orienting_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_and_viewing_plane_orienting_relationship
  SUBTYPE OF (managed_design_object);
    orienting_viewing_plane : viewing_plane;
    oriented_tolerance_zone : tolerance_zone;
  INVERSE
    zone_orientation : viewing_plane_based_orientation FOR orienting_relationship;
END_ENTITY; -- tolerance_zone_and_viewing_plane_orienting_relationship

( *
```

4.2.915 Tolerance_zone_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_boundary is a type of Non_feature_shape_element (see 4.2.560) that is a boundary of a Tolerance_zone (see 4.2.913). A Tolerance_zone_boundary is one of a Conical_tolerance_zone_boundary (see 4.2.178), a Tolerance_zone_circular_or_cylindrical_or_spherical_boundary (see 4.2.919), or a Tolerance_zone_opposing_boundary_set (see 4.2.926).

EXAMPLE 1 A pair of opposing planes may form a Tolerance_zone_boundary.

The data associated with a Tolerance_zone_boundary are the following:

- boundary_zone_definition;
- bounded_zone.

4.2.915.1 boundary_zone_definition

The boundary_zone_definition is an inverse relationship that specifies that the existence of the Tolerance_zone_boundary is dependent on the existence of the Tolerance_zone_boundary_definition (see 4.2.916) that specifies the Tolerance_zone_boundary as its defined_zone_boundary.

4.2.915.2 bounded_zone

The bounded_zone is an inverse relationship that specifies that the existence of the Tolerance_zone_boundary is dependent on the existence of the Tolerance_zone (see 4.2.913) that specifies the Tolerance_zone_boundary as its zone_boundary.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_boundary
  ABSTRACT SUPERTYPE OF (ONEOF (
    tolerance_zone_circular_or_cylindrical_or_spherical_boundary,
    tolerance_zone_opposing_boundary_set,
    conical_tolerance_zone_boundary))
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    bounded_zone          : tolerance_zone FOR zone_boundary;
    boundary_zone_definition : tolerance_zone_boundary_definition FOR
                                defined_zone_boundary;
END_ENTITY; -- tolerance_zone_boundary

( *
```

4.2.916 Tolerance_zone_boundary_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_boundary_definition is a type of Managed_design_object (see 4.2.524) that characterizes a Tolerance_zone_boundary (see 4.2.915). A Tolerance_zone_boundary_definition is either a Boundary_zone_definition_with_specified_size (see 4.2.79) or a Conical_tolerance_zone_boundary_definition (see 4.2.180).

The data associated with a Tolerance_zone_boundary_definition are the following:

- defined_zone_boundary;
- zone_definition.

4.2.916.1 defined_zone_boundary

The defined_zone_boundary specifies the Tolerance_zone_boundary (see 4.2.915) that is characterized by the Tolerance_zone_boundary_definition.

4.2.916.2 zone_definition

The zone_definition is an inverse relationship that specifies that the existence of the Tolerance_zone_boundary_definition is dependent on the existence of the Tolerance_zone_definition (see 4.2.920) that specifies the Tolerance_zone_boundary_definition as its boundary_definition.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_boundary_definition
  ABSTRACT SUPERTYPE OF (ONEOF (
    boundary_zone_definition_with_specified_size,
    conical_tolerance_zone_boundary_definition))
  SUBTYPE OF (managed_design_object);
  defined_zone_boundary : tolerance_zone_boundary;
  INVERSE
    zone_definition : tolerance_zone_definition FOR boundary_definition;
END_ENTITY; -- tolerance_zone_boundary_definition

( *
```

4.2.917 Tolerance_zone_boundary_member

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_boundary_member is a type of Non_feature_shape_element (see 4.2.560) that is one of the two explicitly identified members of a Tolerance_zone_explicit_opposing_boundary_set (see 4.2.923).

The data associated with a Tolerance_zone_boundary_member are the following:

— explicit_opposing_boundary_set.

4.2.917.1 explicit_opposing_boundary_set

The explicit_opposing_boundary_set is an inverse relationship that specifies that the existence of the Tolerance_zone_boundary_member is dependent on the existence of the Tolerance_zone_explicit_opposing_boundary_set (see 4.2.923) that specifies the Tolerance_zone_boundary_member as one of its boundary_members.

EXAMPLE 1 Tolerance_zone_boundary_members may be used to specify an unequally disposed profile tolerance (i.e., either a Profile_of_any_line_tolerance (see 4.2.744) or Profile_of_any_surface_tolerance).

Constraints:

WR1: The `Tolerance_zone_boundary_member` shall not be specified as the `offset_tolerance_zone_boundary_member` by more than one `Tolerance_zone_boundary_member_offset`.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_boundary_member
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    explicit_opposing_boundary_set :
    tolerance_zone_explicit_opposing_boundary_set FOR boundary_members;
END_ENTITY; -- tolerance_zone_boundary_member

( *
```

4.2.918 Tolerance_zone_boundary_member_offset

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tolerance_zone_boundary_member_offset` is a type of `Managed_design_object` (see 4.2.524). A `Tolerance_zone_boundary_member_offset` is a definition that characterizes the size and direction that a `Tolerance_zone_boundary_member` (see 4.2.917) is offset from the `True_profile` (see 4.2.934) of the tolerated `Physical_feature_or_part_template` (see 4.2.658).

The data associated with a `Tolerance_zone_boundary_member_offset` are the following:

- `boundary_definition`;
- `offset_tolerance_zone_boundary_member`;
- `offset_value`;
- `type_of_offset`.

4.2.918.1 boundary_definition

The `boundary_definition` is an inverse relationship that specifies that the existence of the `Tolerance_zone_boundary_member_offset` is dependent on the existence of the `Profile_boundary_definition_with_offsets` (see 4.2.743) that specifies the `Tolerance_zone_boundary_member_offset` as one of its `boundary_offsets`.

4.2.918.2 offset_tolerance_zone_boundary_member

The offset_tolerance_zone_boundary_member specifies the Tolerance_zone_boundary_member (see 4.2.917) the offset of which is characterized by the Tolerance_zone_boundary_member_offset.

4.2.918.3 offset_value

The offset_value specifies the Length_data_element (see 4.2.511) that represents distance the Tolerance_zone_boundary_member (see 4.2.917) is offset from the True_profile (see 4.2.934).

4.2.918.4 type_of_offset

The type_of_offset specifies a label that indicates the direction that the Tolerance_zone_boundary_member (see 4.2.917) is offset from the True_profile (see 4.2.934). The type_of_offset shall either be "away_from_material" or "towards_material".

Constraints:

WR1: The Tolerance_zone_explicit_opposing_boundary_set that is specified as the explicit_opposing_boundary_set by the Tolerance_zone_boundary_member that is specified as the offset_tolerance_zone_boundary_member shall be the same instance of Tolerance_zone_explicit_opposing_boundary_set that is specified as the defined_zone_boundary by the Profile_boundary_definition_with_offsets specified as the boundary_definition.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_boundary_member_offset
  SUBTYPE OF (managed_design_object);
    offset_tolerance_zone_boundary_member : tolerance_zone_boundary_member;
    offset_value                          : length_data_element;
    type_of_offset                        : offset_types;
  INVERSE
    boundary_definition : profile_boundary_definition_with_offsets FOR
                          boundary_offsets;
END_ENTITY; -- tolerance_zone_boundary_member_offset

( *
```

4.2.919 Tolerance_zone_circular_or_cylindrical_or_spherical_boundary

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tolerance_zone_circular_or_cylindrical_or_spherical_boundary` is a type of `Tolerance_zone_boundary` (see 4.2.915) that is circular, cylindrical, or spherical in form. The actual form of the `Tolerance_zone_boundary` is dictated by the type of tolerance and the form of the tolerated `Shape_element`.

The data associated with a `Tolerance_zone_circular_or_cylindrical_or_spherical_boundary` are the following:

— `size_characteristic`.

4.2.919.1 `size_characteristic`

A `size_characteristic` is an inverse relationship that specifies that the existence of the `Tolerance_zone_circular_or_cylindrical_or_spherical_boundary` is dependent on the existence of the `Diametrical_boundary_size_characteristic` (see 4.2.287) that specifies the `Tolerance_zone_circular_or_cylindrical_or_spherical_boundary` as its `sized_boundary`.

EXAMPLE 1 A cylindrical `Tolerance_zone_boundary` (see 4.2.915) could be defined by a position tolerance applied to an axis of a cylindrical hole.

NOTE EXPRESS specification:

```
*)
ENTITY tolerance_zone_circular_or_cylindrical_or_spherical_boundary
  SUBTYPE OF (tolerance_zone_boundary);
  INVERSE
    size_characteristic : diametrical_boundary_size_characteristic FOR
                          sized_boundary;
END_ENTITY; -- tolerance_zone_circular_or_cylindrical_or_spherical_boundary

( *
```

4.2.920 `Tolerance_zone_definition`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Tolerance_zone_definition` is a type of `Managed_design_object` (see 4.2.524). A `Tolerance_zone_definition` may be a one of `Conical_tolerance_zone_definition` (see 4.2.182), `Projected_tolerance_zone_definition` (see 4.2.752), `Runout_tolerance_zone_definition_with_specified_angle` (see 4.2.796), or `Tolerance_zone_definition_with_per_unit_size_specification` (see 4.2.921). Additionally, a `Tolerance_zone_definition` may be a `Tolerance_zone_definition_with_specified_orientation` (see 4.2.922). A `Tolerance_zone_definition` is a definition that characterizes a `Tolerance_zone` (see 4.2.913).

The data associated with a Tolerance_zone_definition are the following:

- boundary_definition;
- defined_tolerance_zone;
- form.

4.2.920.1 boundary_definition

The boundary_definition specify the Tolerance_zone_boundary_definition (see 4.2.916) that defines the limits of the Tolerance_zone (see 4.2.913) for the Tolerance_zone_definition.

4.2.920.2 defined_tolerance_zone

The defined_tolerance_zone specifies the Tolerance_zone (see 4.2.913) that is characterized by the Tolerance_zone_definition.

4.2.920.3 form

The form specifies the Tolerance_zone_form (see 4.2.924) that characterizes the form of the specified Tolerance_zone (see 4.2.913). The form need not be specified for a particular Tolerance_zone_definition.

Constraints:

WR1: If the Tolerance_zone_form is specified, the Tolerance_zone specified as the characterized_tolerance_zone of the Tolerance_zone_form that is specified as the form shall be the same instance of Tolerance_zone as specified as the defined_tolerance_zone.

WR2: The Tolerance_zone_boundary specified as the defined_zone_boundary of the Tolerance_zone_boundary_definition that is specified as the boundary_definition shall be the same instances of Tolerance_zone_boundary as specified as the zone_boundary of the Tolerance_zone specified as the defined_tolerance_zone.

WR3: If the Tolerance_zone_definition is not a Conical_tolerance_zone_definition, the Tolerance_zone_definition shall be specified as the tolerance_zone_or_boundary_definition by exactly one Geometric_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_definition
  SUPERTYPE OF (ONEOF (projected_tolerance_zone_definition,
    runout_tolerance_zone_definition_with_specified_angle,
    tolerance_zone_definition_with_per_unit_size_specification,
    conical_tolerance_zone_definition,
    tolerance_zone_definition_with_specified_orientation))
  SUBTYPE OF (managed_design_object);
  defined_tolerance_zone : tolerance_zone;
  boundary_definition    : tolerance_zone_boundary_definition;
  form                   : OPTIONAL tolerance_zone_form;
END_ENTITY; -- tolerance_zone_definition

( *

```

4.2.921 Tolerance_zone_definition_with_per_unit_size_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_with_per_unit_size_specification is a type of Tolerance_zone_definition (see 4.2.920) in which the extent of the Tolerance_zone (see 4.2.913) is defined. A Tolerance_zone_with_per_unit_size_specification is used to define a Tolerance_zone when the design requirements are such that abrupt variations in the toleranced Shape_element (see 4.2.810) must be limited. In essence, the thickness, or the diameter, of the Tolerance_zone is defined by the Boundary_size_characteristic (see 4.2.78) and the extent is defined by the Tolerance_zone_per_unit_size_specification (see 4.2.927). Each element of the toleranced Shape_element that has the same extent as the Tolerance_zone must lie within the Tolerance_zone when the Tolerance_zone is positioned coincident with the element.

The data associated with a Tolerance_zone_definition_with_per_unit_size_specification are the following:

— per_unit_size_specification.

4.2.921.1 per_unit_size_specification

The per_unit_size_specification specifies the Tolerance_zone_per_unit_size_specification (see 4.2.927) that defines the extent of the specified Tolerance_zone (see 4.2.913) for the Tolerance_zone_definition_with_per_unit_size_specification.

EXAMPLE 1 A Straightness_tolerance may be applied on a per unit basis such that every 100 mm length of a toleranced axis must lie within a Tolerance_zone that is bounded by a cylindrical Tolerance_zone that is 3 mm in diameter.

Constraints:

WR1: The Tolerance_zone specified as the defined_tolerance_zone shall be the same instance of Tolerance_zone as specified as the specified_tolerance_zone by the Tolerance_zone_per_unit_size_specification that is specified as the per_unit_size_specification.

NOTE The attribute defined_tolerance_zone is declared in the Tolerance_zone_definition entity of which this entity is a subtype.

WR2: The Geometric_tolerance that specifies the Tolerance_zone_definition_with_per_unit_size_specification as the tolerance_zone_or_boundary_definition shall be one of Flatness_tolerance, Straightness_tolerance, Angularity_tolerance, Parallelism_tolerance, or Perpendicularity_tolerance.

NOTE EXPRESS specification:

```
*)
ENTITY tolerance_zone_definition_with_per_unit_size_specification
  SUBTYPE OF (tolerance_zone_definition);
    per_unit_size_specification : tolerance_zone_per_unit_size_specification;
END_ENTITY; -- tolerance_zone_definition_with_per_unit_size_specification

( *
```

4.2.922 Tolerance_zone_definition_with_specified_orientation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_definition_with_specified_orientation is a type of Tolerance_zone_definition (see 4.2.920) in which the orientation of the Tolerance_zone (see 4.2.913) is specified.

The data associated with a Tolerance_zone_definition_with_specified_orientation are the following:

— zone_orientation.

4.2.922.1 zone_orientation

The zone_orientation specifies either a Viewing_plane_based_orientation (see 4.2.957), a Datum_axis_related_orientation (see 4.2.217), or a Datum_based_vector_orientation (see 4.2.220) for the Tolerance_zone_definition_with_specified_orientation.

Constraints:

WR1: If a Datum_axis_related_orientation is specified as the zone_orientation, the Tolerance_zone specified as the defined_tolerance_zone shall be the same instance of Tolerance_zone as specified as the oriented_tolerance_zone by the Datum_axis_and_tolerance_zone_orienting_relationship that is specified as the orienting_relationship by the Datum_axis_related_orientation that is specified as the zone_orientation.

WR2: If a Viewing_plane_based_orientation is specified as the zone_orientation, the Tolerance_zone specified as the defined_tolerance_zone shall be the same instance of Tolerance_zone as specified as the oriented_tolerance_zone by the Tolerance_zone_and_viewing_plane_orienting_relationship that is specified as the orienting_relationship by the Viewing_plane_based_orientation that is specified as the zone_orientation.

WR3: The Tolerance_zone_definition_with_specified_orientation shall not be specified as a tolerance_zone_or_boundary_definition by either a Circular_runout_tolerance, a Circularity_tolerance, a Concentricity_and_coaxiality_tolerance, a Cylindricity_tolerance, a Flatness_tolerance, a Profile_of_any_surface_tolerance, a Symmetry_tolerance, or a Total_runout_tolerance.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_definition_with_specified_orientation
  SUBTYPE OF (tolerance_zone_definition);
    zone_orientation : orientation_select;
END_ENTITY; -- tolerance_zone_definition_with_specified_orientation

( *
```

4.2.923 Tolerance_zone_explicit_opposing_boundary_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_explicit_opposing_boundary_set is a type of Tolerance_zone_opposing_boundary_set (see 4.2.926) in which the two opposing boundary members are explicitly identified.

The data associated with a Tolerance_zone_explicit_opposing_boundary_set are the following:

— boundary_members.

4.2.923.1 boundary_members

The boundary_members specify the Tolerance_zone_boundary_member (see 4.2.917) that comprise the Tolerance_zone_explicit_opposing_boundary_set. There shall be exactly two Tolerance_zone_boundary_members specified as boundary_members for a Tolerance_zone_explicit_opposing_boundary_set.

NOTE EXPRESS specification:

```
*)
ENTITY tolerance_zone_explicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_opposing_boundary_set);
  boundary_members : SET [2:2] OF tolerance_zone_boundary_member;
END_ENTITY; -- tolerance_zone_explicit_opposing_boundary_set

( *
```

4.2.924 Tolerance_zone_form

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_form is a type of Managed_design_object (see 4.2.524). A Tolerance_zone_form is a definition that characterizes of the form of the Tolerance_zone (see 4.2.913).

The data associated with a Tolerance_zone_form are the following:

- characterized_tolerance_zone;
- name;
- zone_definition.

4.2.924.1 characterized_tolerance_zone

The characterized_tolerance_zone specifies the Tolerance_zone (see 4.2.913) for which the Tolerance_zone_form is defined.

4.2.924.2 name

The name specifies the identifying label of the Tolerance_zone_form.

EXAMPLE 1 The name of the Tolerance_zone_form might be "cylindrical" for a Tolerance_zone that is defined by a Perpendicularity_tolerance.

4.2.924.3 zone_definition

The zone_definition is an inverse relationship that specifies that the existence of the Tolerance_zone_form is dependent on the existence of the Tolerance_zone_definition (see 4.2.920) that specifies the Tolerance_zone_form as its form.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_form
  SUBTYPE OF (managed_design_object);
    characterized_tolerance_zone : tolerance_zone;
    name                          : label;
  INVERSE
    zone_definition : tolerance_zone_definition FOR form;
END_ENTITY; -- tolerance_zone_form

```

(*

4.2.925 Tolerance_zone_implicit_opposing_boundary_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_implicit_opposing_boundary_set is a type of Tolerance_zone_opposing_boundary_set (see 4.2.926) in which the two opposing boundary members are not explicitly identified.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_implicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_opposing_boundary_set);
END_ENTITY; -- tolerance_zone_implicit_opposing_boundary_set

```

(*

4.2.926 Tolerance_zone_opposing_boundary_set

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_opposing_boundary_set is a type of Tolerance_zone_boundary (see 4.2.915) that consists of a set of two like boundary members that are separated by a constant distance. A Tolerance_zone_opposing_boundary_set is either a Tolerance_zone_explicit_opposing_boundary_set (see 4.2.923) or a Tolerance_zone_implicit_opposing_boundary_set (see 4.2.925).

The data associated with a Tolerance_zone_opposing_boundary_set are the following:

— size_characteristic.

4.2.926.1 size_characteristic

The size_characteristic is an inverse relationship that specifies that the existence of the Tolerance_zone_opposing_boundary_set is dependent on the existence of the Opposing_boundary_set_size_characteristic (see 4.2.568) that specifies the Tolerance_zone_opposing_boundary_set as its sized_boundary_set.

EXAMPLE 1 Two opposing planes comprise the Tolerance_zone_boundary_set that is defined when a Flatness_tolerance (see 4.2.386) is applied to the top surface of a printed circuit board. The size_characteristic specifies the separation between the two opposing planes.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_opposing_boundary_set
  ABSTRACT SUPERTYPE OF (ONEOF (
    tolerance_zone_explicit_opposing_boundary_set,
    tolerance_zone_implicit_opposing_boundary_set))
  SUBTYPE OF (tolerance_zone_boundary);
  INVERSE
    size_characteristic : opposing_boundary_set_size_characteristic FOR
                        sized_boundary_set;
END_ENTITY; -- tolerance_zone_opposing_boundary_set

( *
```

4.2.927 Tolerance_zone_per_unit_size_specification

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Tolerance_zone_per_unit_size_specification is a type of Managed_design_object (see 4.2.524). A Tolerance_zone_per_unit_size_specification is a specification of the Tolerance_zone_boundary (see 4.2.915) that defines the extent of the Tolerance_zone_boundary.

The data associated with a Tolerance_zone_per_unit_size_specification are the following:

- per_unit_size ;
- specified_tolerance_zone;
- zone_definition.

4.2.927.1 per_unit_size

The per_unit_size specifies the role of the Length_data_element (see 4.2.511) that represents the extent of the Tolerance_zone (see 4.2.913) for the Tolerance_zone_per_unit_size_specification. There may be one or two Length_data_elements specified as the per_unit_size for the Tolerance_zone_per_unit_size_specification. If one Length_data_element is specified, it represents the linear extent of the Tolerance_zone. If two Length_data_elements are specified, they represent the rectangular extent of the Tolerance_zone.

4.2.927.2 specified_tolerance_zone

The specified_tolerance_zone specifies the Tolerance_zone (see 4.2.913) the extent of which is defined by the Tolerance_zone_per_unit_size_specification.

4.2.927.3 zone_definition

The zone_definition is an inverse relationship that specifies that the existence of the Tolerance_zone_per_unit_size_specification is dependent on the existence of the Tolerance_zone_definition_with_per_unit_size_specification (see 4.2.921) that specifies the Tolerance_zone_per_unit_size_specification as its per_unit_size_specification.

NOTE EXPRESS specification:

```

*)
ENTITY tolerance_zone_per_unit_size_specification
  SUBTYPE OF (managed_design_object);
    specified_tolerance_zone : tolerance_zone;
    per_unit_size            : SET [1:2] OF length_data_element;
  INVERSE
    zone_definition : tolerance_zone_definition_with_per_unit_size_specification FOR
                      per_unit_size_specification;
END_ENTITY; -- tolerance_zone_per_unit_size_specification

( *
```

4.2.928 Topological_junction

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Topological_junction is a type of Managed_design_object (see 4.2.524). A Topological_junction is used when at least three Component_terminal (see 4.2.137) are connected and the explicit topology must be stated.

The data associated with a Topological_junction are the following:

- name;
- scope.

4.2.928.1 name

The name identifies the Topological_junction.

4.2.928.2 scope

The scope specifies the role of the Physical_connectivity_structure_definition (see 4.2.657) for the Topological_junction.

Constraints:

UR1: The combination of the name and the scope shall be unique within a population of Topological_junction.

NOTE EXPRESS specification:

```
*)
ENTITY topological_junction
  SUBTYPE OF (managed_design_object);
    scope : physical_connectivity_structure_definition;
    name  : ee_name;
  UNIQUE
    UR1 : name, scope;
END_ENTITY; -- topological_junction

( *
```

4.2.929 Total_runout_tolerance

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Total_runout_tolerance` is a type of `Geometric_tolerance_with_referenced_datum_system` (see 4.2.422) that specifies or partially specifies the design requirement on the run-out of a `Physical_feature_or_part_template` (see 4.2.658) in relation to a `Datum_axis` (see 4.2.215) of a `Datum_system` (see 4.2.236). Where run-out is the quality or state of having perfect form and axial alignment in relation to an axis. A `Total_runout_tolerance` may either be applied normal to the surface or at a specified angle with respect to the `Datum_axis`. If the `Total_runout_tolerance` is applied at a specified angle the `Tolerance_zone_definition` (see 4.2.920) is a `Runout_tolerance_zone_definition_with_specified_angle` (see 4.2.796). The `Physical_feature_or_part_template` may be either coaxial or normal to the `Datum_axis`.

NOTE Total run-out tolerance is defined in terms of tolerance zone definitions in 14.14 of ISO 1101.

Constraints:

WR1: The `Shape_element` specified as the `toleranced_shape_element` shall be a `Physical_feature_or_part_template`.

NOTE The attribute `toleranced_shape_element` is declared in the `Geometric_tolerance` entity of which this entity is a subtype.

WR2: The `Total_runout_tolerance` shall not be a `Geometric_tolerance_with_applied_material_condition_property`.

WR3: There shall be at most two `Datum` in the `Datum_system` the definition of which is specified by the `datum_system_definition`.

NOTE The attribute `datum_system_definition` is declared in the `Geometric_tolerance_with_referenced_datum_system` entity of which this entity is a subtype.

WR4: There shall be a `Datum_axis` in the `Datum_system` the definition of which is specified by the `datum_system_definition`.

WR5: The `Datum_system_defining_conditions` that is specified as the `datum_system_definition` shall be a `Datum_system_defining_conditions_without_material_conditions`.

WR6: The `Total_runout_tolerance` shall specify a `Tolerance_zone_definition` as its `tolerance_zone_or_boundary_definition`; and that `Tolerance_zone_definition` shall specify a `Tolerance_zone_boundary_definition_with_specified_size` as its `boundary_definition`; and that `Tolerance_zone_boundary_definition_with_specified_size` shall specify an `Opposing_boundary_set_size_characteristic` as its `specified_boundary_size`.

IP1: The form of the `toleranced Physical_feature_or_part_template` shall be consistent with the form of a physical feature for which the application of a total run-out tolerance is sensible.

NOTE EXPRESS specification:

```
*)
ENTITY total_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- total_runout_tolerance

( *
```

4.2.930 Trace_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Trace_template is a type of Part_template (see 4.2.620) whose purpose is to define the width of a Conductive_interconnect_element (see 4.2.171) and the style of the ends and corners of that Conductive_interconnect_element. A Trace_template may be a Default_trace_template (see 4.2.252). A Trace_template shall not specify a curve_font on the Curve_style (see 4.2.205) specified by trace_style.

EXAMPLE 1 Figure 110 illustrates the cross-section view related to a Trace_template. The parameters included in the illustration are included in the Model_parameter (see 4.2.549) for the associated Analytical_model (see 4.2.14) and will be assigned specific values through Parameter_assignment (see 4.2.603) and associated Analytical_representation (see 4.2.22) application objects.

The data associated with a Trace_template are the following:

— trace_style.

4.2.930.1 trace_style

The trace_style specifies the role of the Curve_style for the Trace_template.

NOTE The Trace_template uses the parametric information in Curve_style, not the graphical presentation information.

NOTE EXPRESS specification:

```
*)
ENTITY trace_template
  SUPERTYPE OF (default_trace_template)
  SUBTYPE OF (part_template);
  trace_style : curve_style;
END_ENTITY; -- trace_template

( *
```

4.2.931 Transmission_line_functional_unit_terminal_allocation

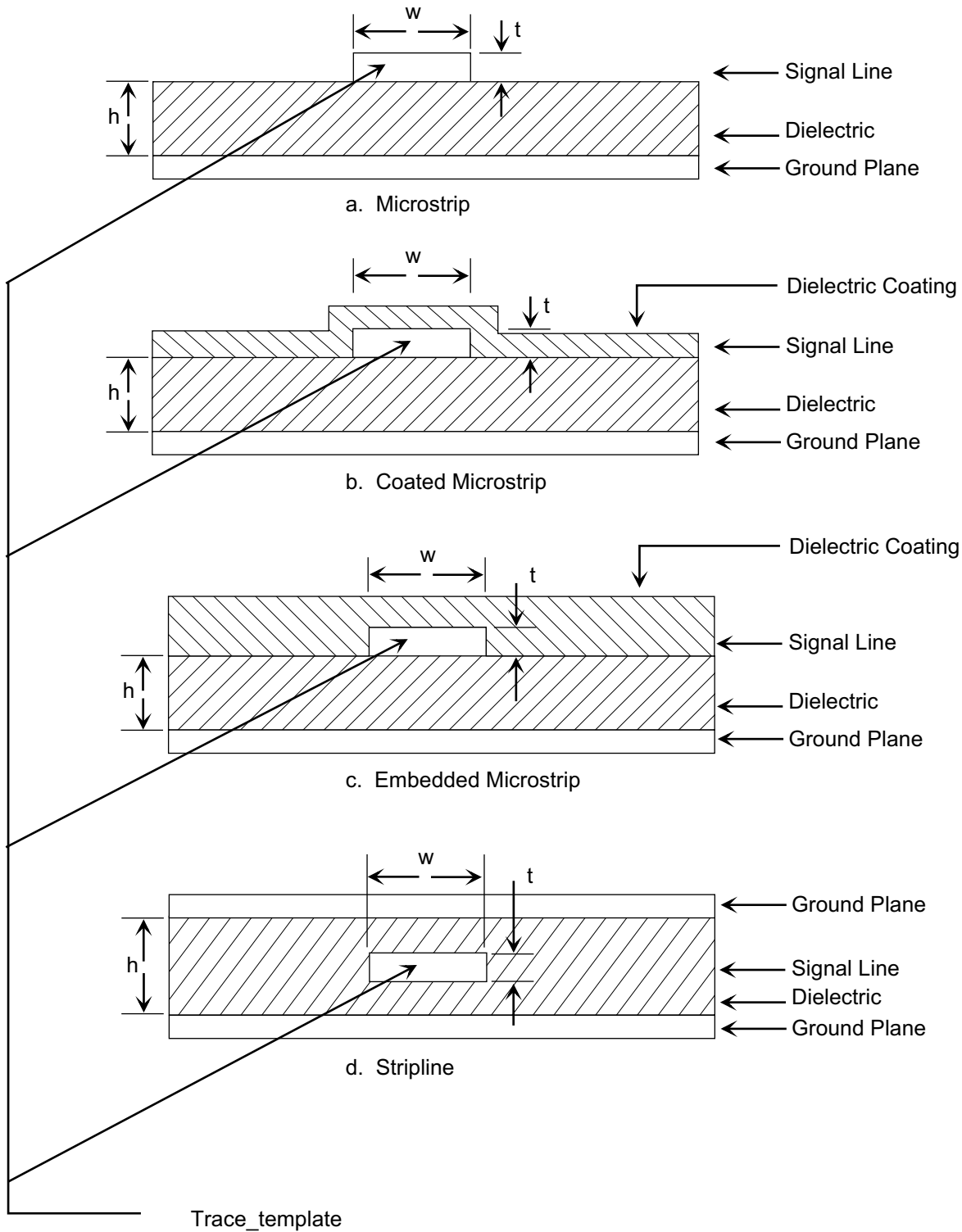


Figure 110 – Trace_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Transmission_line_functional_unit_terminal_allocation` is a type of `Managed_design_object` (see 4.2.524).
A `Transmission_line_functional_unit_terminal_allocation` is the relationship between the occurrence of a `Functional_unit_terminal` (see 4.2.407) of a transmission line and the physical implementation of that occurrence as a `Printed_component_join_terminal` (see 4.2.722).

The data associated with a `Transmission_line_functional_unit_terminal_allocation` are the following:

- `functional_unit_design_path`;
- `implementation`;
- `transmission_line_functional_unit_terminal`.

4.2.931.1 functional_unit_design_path

The `functional_unit_design_path` specifies the `Transmission_line_functional_unit_to_routed_printed_component_allocation` (see 4.2.932) path from the `Functional_unit_terminal` (see 4.2.407) to the `Printed_component_join_terminal` (see 4.2.722) for the `Transmission_line_functional_unit_terminal_allocation`.

4.2.931.2 implementation

The `implementation` specifies the role of the `Printed_component_join_terminal` (see 4.2.722) for the `Transmission_line_functional_unit_terminal_allocation`.

4.2.931.3 transmission_line_functional_unit_terminal

The `transmission_line_functional_unit_terminal` specifies the role of the `Functional_unit_terminal` (see 4.2.407) for this `Transmission_line_functional_unit_terminal_allocation`.

Constraints:

UR1: The combination of the `implementation` and `transmission_line_functional_unit_terminal` shall be unique within a population of `Transmission_line_functional_unit_terminal_allocation`.

NOTE EXPRESS specification:

```

*)
ENTITY transmission_line_functional_unit_terminal_allocation
  SUBTYPE OF (managed_design_object);
    implementation                : printed_component_join_terminal;
    transmission_line_functional_unit_terminal : functional_unit_terminal;
    functional_unit_design_path    :
transmission_line_functional_unit_to_routed_printed_component_allocation;
  UNIQUE
    UR1 : implementation, transmission_line_functional_unit_terminal;
END_ENTITY; -- transmission_line_functional_unit_terminal_allocation

( *

```

4.2.932 Transmission_line_functional_unit_to_routed_printed_component_allocation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Transmission_line_functional_unit_to_routed_printed_component_allocation is a type of Design_functional_unit_allocation_to_assembly_component (see 4.2.263). A transmission_line_functional_unit_to_routed_printed_component_allocation is the allocation of the functional requirements specified by a member of Functional_unit_definition (see 4.2.401) in a hierarchical definition to the Conductive_interconnect_element (see 4.2.171).

The data associated with a Transmission_line_functional_unit_to_routed_printed_component_allocation are the following:

— implementation.

4.2.932.1 implementation

The implementation specifies the role of the Routed_printed_component (see 4.2.779) for the Transmission_line_functional_unit_to_routed_printed_component_allocation.

NOTE EXPRESS specification:

```

*)
ENTITY transmission_line_functional_unit_to_routed_printed_component_allocation
  SUBTYPE OF (design_functional_unit_allocation_to_assembly_component);
  SELF\design_functional_unit_allocation_to_assembly_component.implementation :
  routed_printed_component;
END_ENTITY;
-- transmission_line_functional_unit_to_routed_printed_component_allocation

--tt tc1 wd12 modified
--tt tc1 wd9 modified
( *
```

4.2.933 Trimmed_line

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Trimmed_line is a type of Curve (see 4.2.203) that has a defined, finite extent. The Trimmed_line shall be interpreted as a trimmed curve.

NOTE The Trimmed_line is necessary to support the specification of precise, metric geometry.

NOTE EXPRESS specification:

```

*)
ENTITY trimmed_line
  SUBTYPE OF (curve);
  start_point : cartesian_point;
  end_point : cartesian_point;
END_ENTITY; -- trimmed_line

( *
```

4.2.934 True_profile

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A True_profile is a type of Managed_design_object (see 4.2.524). A True_profile is a definition that characterizes the theoretically exact shape of a Physical_feature_or_part_template (see 4.2.658).

The data associated with a True_profile are the following:

- boundary_definition;
- profile_representation;
- profiled_surface.

4.2.934.1 boundary_definition

The boundary_definition is an inverse relationship that specifies that the existence of the True_profile is dependent on the existence of the Profile_boundary_definition (see 4.2.742) that specifies the True_profile as its basic_profile. There shall be one or more Profile_boundary_definition for a True_profile.

4.2.934.2 profile_representation

The profile_representation specifies one of a Bound_volume_shape (see 4.2.76), a Dimension_set (see 4.2.297), an Ee_document (see 4.2.316), or an Ee_rule_function (see 4.2.334) that represents the True_profile.

4.2.934.3 profiled_surface

The profiled_surface specifies the Physical_feature_or_part_template (see 4.2.658) that is characterized by the True_profile

NOTE EXPRESS specification:

```

*)
ENTITY true_profile
  SUBTYPE OF (managed_design_object);
  profiled_surface      : physical_feature_or_part_template;
  profile_representation : true_profile_representations;
  INVERSE
    boundary_definition : SET [1:?] OF profile_boundary_definition FOR
                                basic_profile;
END_ENTITY; -- true_profile

( *
```

4.2.935 Type_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Type_qualifier is a type of Managed_design_object (see 4.2.524) which specifies an uncertainty. The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY type_qualifier
  SUBTYPE OF (managed_design_object);
  name : label;
END_ENTITY;
(*
```

4.2.936 Uncertainty_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Uncertainty_qualifier is a type of Managed_design_object (see 4.2.524). The uncertainty may be Standard_uncertainty or Qualitative_uncertainty. The uncertainty is defined in clause 2 of "The Guide to the Expression of Uncertainty in Measurement".

NOTE EXPRESS specification:

```
*)
ENTITY uncertainty_qualifier
  SUPERTYPE OF (ONEOF(standard_uncertainty, qualitative_uncertainty))
  SUBTYPE OF (managed_design_object);
  measure_name : text;
  description : text;
END_ENTITY;
(*
```

4.2.937 Unrouted_conductive_interconnect_element

An Unrouted_conductive_interconnect_element is a type of Unrouted_conductive_interconnect_element (see 4.2.937). The data provided by Unrouted_conductive_interconnect_element is provided to identify an unrouted portion of a Physical_network (see 4.2.660).

The data associated with an Unrouted_conductive_interconnect_element are the following:

— connected_point.

4.2.937.1 connected_point

The connected_point specifies the role of the Layer_connection_point_link (see 4.2.500) for the Unrouted_conductive_interconnect_element. There shall be one or more Layer_connection_point_link for an Unrouted_conductive_interconnect_element. No Layer_connection_point_link shall be specified more than once as a connected_point for an Unrouted_conductive_interconnect_element.

NOTE EXPRESS specification:

```

*)
ENTITY unrouted_conductive_interconnect_element
  SUBTYPE OF (conductive_interconnect_element);
  connected_points : SET [1:?] OF layer_connection_point_link;
WHERE
  WR1: NOT(EXISTS(SELF\conductive_interconnect_element.composed_conductor));
END_ENTITY; -- unrouted_conductive_interconnect_element
( *

```

4.2.938 Unrouted_join_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Unrouted_join_relationship is a type of Join_relationship (see 4.2.479), in which the terminals referenced, Laminate_component_join_terminal (see 4.2.486), have not been converted into another type of Join_relationship. An Unrouted_join_relationship is an identification of items that are unrouted in a Physical_network.

NOTE EXPRESS specification:

```

*)
ENTITY unrouted_join_relationship
  SUBTYPE OF (join_relationship);
  unrouted_terminals : SET [2:?] OF laminate_component_join_terminal;
END_ENTITY; -- unrouted_join_relationship
( *

```

4.2.939 Unsupported_passage

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Unsupported_passage is a type of Inter_stratum_feature (see 4.2.432) that has no metallization in the realization of its walls.

The data associated with an Unsupported_passage are the following:

— usage_definition.

4.2.939.1 usage_definition

The usage_definition specifies the role of the Unsupported_passage_template (see 4.2.941) for the Unsupported_passage.

Constraints:

WR1: An Unsupported_passage shall be an inner feature.

NOTE EXPRESS specification:

```
*)  
ENTITY unsupported_passage  
    SUBTYPE OF (inter_stratum_feature);  
    SELF\assembly_component.usage_definition : unsupported_passage_template;  
END_ENTITY; -- unsupported_passage  
  
(*
```

4.2.940 Unsupported_passage_dependent_non_functional_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Unsupported_passage_dependent_non_functional_land is a type of Non_functional_land (see 4.2.562) whose shape is dependent upon the shape of an Unsupported_passage.

The data associated with an Unsupported_passage_dependent_non_functional_land are the following:

- reference_passage;
- usage_definition.

4.2.940.1 reference_passage

The reference_passage specifies the role of the Unsupported_passage (see 4.2.939) for the Unsupported_passage_dependent_non_functional_land.

4.2.940.2 usage_definition

The usage_definition specifies the role of the Default_unsupported_passage_based_land_physical_template (see 4.2.253) for the Unsupported_passage_dependent_non_functional_land.

NOTE EXPRESS specification:

```

*)
ENTITY unsupported_passage_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
    reference_passage : unsupported_passage;
    SELF\assembly_component.usage_definition :
default_unsupported_passage_based_land_physical_template;
END_ENTITY; -- unsupported_passage_dependent_non_functional_land

( *

```

4.2.941 Unsupported_passage_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

An Unsupported_passage_template is a type of Inter_stratum_feature_template (see 4.2.435).

NOTE The Unsupported_passage_template associates shape information with the passage_technology parametric property information. "Unsupported" carries the meaning that the passage is not mechanically or electrically supported by deposited sidewalls.

The data associated with an Unsupported_passage_template are the following:

— of_passage_technology.

4.2.941.1 of_passage_technology

The of_passage_technology specifies the role of the Default_unsupported_passage_definition (see 4.2.254) for the Unsupported_passage_template.

NOTE EXPRESS specification:

```

*)
ENTITY unsupported_passage_template
  SUPERTYPE OF (ONEOF (counterbore_passage_template,
    countersunk_passage_template, blind_passage_template))
  SUBTYPE OF (inter_stratum_feature_template);
    SELF\inter_stratum_feature_template.of_passage_technology :
default_unsupported_passage_definition;
END_ENTITY; -- unsupported_passage_template

( *

```

4.2.942 Usage_concept_usage_relationship

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Usage_concept_usage_relationship` is a type of `Managed_design_object` (see 4.2.524). A `Usage_concept_usage_relationship` is the association between a `Component_termination_passage_template_terminal` (see 4.2.153), a `Land_template_terminal` (see 4.2.496), a `Part_feature` (see 4.2.609), a `Part_template` (see 4.2.620), a `Physical_unit_usage_view` (see 4.2.672), a `Printed_part_template_terminal` (see 4.2.737), or a `Via_template_terminal` (see 4.2.953), as a source and a `Component_termination_passage_template_terminal`, a `Land_template_terminal`, a `Part_feature`, a `Part_template`, a `Physical_unit_usage_view`, a `Printed_part_template_terminal` or a `Via_template_terminal` as a target. This relationship positions and orients the geometric model of the source into the model of the target using a transformation.

The data associated with an `Usage_concept_usage_relationship` are the following:

- `associated_usage`;
- `associated_usage_placement`;
- `associating_usage`.

4.2.942.1 `associated_usage`

The `associated_usage` specifies either a `Component_termination_passage_template_terminal` (see 4.2.153), a `Land_template_terminal` (see 4.2.496), a `Part_feature` (see 4.2.609), a `Part_template` (see 4.2.620), a `Physical_unit_usage_view` (see 4.2.672), a `Printed_part_template_terminal` (see 4.2.737), or a `Via_template_terminal` (see 4.2.953) as the target item for the `Usage_concept_usage_relationship`.

4.2.942.2 `associated_usage_placement`

The `associated_usage_placement` specifies the `Axis_placement` (see 4.2.56) of the target item in the coordinate system of the source item for the `Usage_concept_usage_relationship`, including any transformation required.

4.2.942.3 `associating_usage`

The `associating_usage` specifies either a `Component_termination_passage_template_terminal` (see 4.2.153), a `Land_template_terminal` (see 4.2.496), a `Part_feature` (see 4.2.609), a `Part_template` (see 4.2.620), a `Physical_unit_usage_view` (see 4.2.672), a `Printed_part_template_terminal` (see 4.2.737), or a `Via_template_terminal` (see 4.2.953) as the source item for the `Usage_concept_usage_relationship`.

Constraints:

WR1: No member specified by `associated_usage` or by `associating_usage` shall be an ancestor or descendant of itself.

IP1: All necessary geometric model relationships to support this requirement shall be included.

NOTE EXPRESS specification:

```

*)
ENTITY usage_concept_usage_relationship
  SUBTYPE OF (managed_design_object);
    associating_non_feature_usage_shape : OPTIONAL non_feature_shape_definition;
    associating_usage_shape      : OPTIONAL part_template_or_physical_unit_shape_select;
    associated_usage              : usage_concept;
    associated_shape_definition : shape_definition;
    associated_usage_placement : axis_placement_or_cto2d;
WHERE
  WR1: (EXISTS (associating_non_feature_usage_shape)) XOR
        (EXISTS (associating_usage_shape));
  WR2: NOT(EXISTS (associating_non_feature_usage_shape)) OR
        ('AP210_ARM.NON_FEATURE_SHAPE_DEFINITION'
         IN TYPEOF(associated_shape_definition));
  WR3: NOT(EXISTS (associating_usage_shape)) OR
        ('AP210_ARM.FEATURE_SHAPE_DEFINITION'
         IN TYPEOF(associated_shape_definition));
  WR4: NOT('AP210_ARM.CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN
          TYPEOF(associated_usage_placement)) OR
        NOT(EXISTS(associating_usage_shape));
END_ENTITY; -- usage_concept_usage_relationship
( *

```

4.2.943 Usage_view_connection_zone_terminal_shape_relationship

A Usage_view_connection_zone_terminal_shape_relationship is a type of Managed_design_object (see 4.2.524). This relationship positions and orients the geometric model of a Connection_zone into the model of feature using a transformation. The interpretation of the associated_connection_zone_shape_definition shall be that it shall be a shape_representation for the Connection_zone in the role of associated_usage.

The data associated with an Usage_view_connection_zone_terminal_shape_relationship are the following:

- associating_terminal_shape;
- associated_usage;
- associated_connection_zone_shape_definition;
- associated_usage_placement.

4.2.943.1 associating_terminal_shape

The associating_terminal_shape specifies the Feature_shape_definition (see 4.2.372) that is the model shape for the Usage_view_connection_zone_terminal_shape_relationship.

4.2.943.2 associated_usage

The associated_usage specifies the Connection_zone (see 4.2.185) that is the target item for the Usage_view_connection_zone_terminal_shape_relationship.

4.2.943.3 associated_connection_zone_shape_definition

The associated_connection_zone_shape_definition specifies the Connection_zone_shape (see ??) that is the source shape for the Usage_view_connection_zone_terminal_shape_relationship.

4.2.943.4 associated_usage_placement

The associated_usage_placement specifies the Axis_placement (see 4.2.56) that is the target in the coordinate system of the model for the Usage_view_connection_zone_terminal_shape_relationship, including any transformation required.

NOTE EXPRESS specification:

```

*)
ENTITY usage_view_connection_zone_terminal_shape_relationship
  SUBTYPE OF (managed_design_object);
    associating_terminal_shape          : feature_shape_definition;
    associated_usage                     : connection_zone;
    associated_connection_zone_shape_definition : feature_shape_definition;
    associated_usage_placement           : axis_placement;
WHERE
  WR1: associating_terminal_shape :#60;#62;;
    associated_connection_zone_shape_definition;
  END_ENTITY; -- usage_view_connection_zone_terminal_shape_relationship
( *
```

4.2.944 User_defined_dimension_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A User_defined_dimension_qualifier is a type of Dimension_qualifier (see 4.2.294) that requires a more detailed description than those stated for a Pre_defined_dimension_qualifier (see 4.2.707).

The data associated with an User_defined_dimension_qualifier are the following:

- description;
- name.

4.2.944.1 description

The description specifies the text that describes the manner in which the Dimensional_characteristic (see 4.2.298) that are specified as the qualified_dimensions are qualified by the User_defined_dimension_qualifier.

NOTE The attribute qualified_dimensions is declared in the Dimension_qualifier entity of which this entity is a subtype.

4.2.944.2 name

The name specifies the label for the User_defined_dimension_qualifier by which it is referred.

NOTE EXPRESS specification:

```
*)
ENTITY user_defined_dimension_qualifier
  SUBTYPE OF (dimension_qualifier);
  description : text;
  name       : text;
END_ENTITY; -- user_defined_dimension_qualifier

( *
```

4.2.945 User_defined_geometric_tolerance_qualifier

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A User_defined_geometric_tolerance_qualifier is a type of Geometric_tolerance_qualifier (see 4.2.420) that requires a more detailed description than those stated for a Pre_defined_geometric_tolerance_qualifier (see 4.2.708).

The data associated with an User_defined_geometric_tolerance_qualifier are the following:

- description;
- name.

4.2.945.1 description

The description specifies the text that describes the manner in which the Geometric_tolerance (see 4.2.419) that are specified as the qualified_geometric_tolerances are qualified by the User_defined_geometric_tolerance_qualifier.

NOTE The attribute qualified_geometric_tolerances is declared in the Geometric_tolerance_qualifier entity of which this entity is a subtype.

4.2.945.2 name

The name specifies the label for the User_defined_geometric_tolerance_qualifier by which it is referred.

NOTE EXPRESS specification:

```
*)
ENTITY user_defined_geometric_tolerance_qualifier
  SUBTYPE OF (geometric_tolerance_qualifier);
  description : text;
  name       : text;
END_ENTITY; -- user_defined_geometric_tolerance_qualifier

( *
```

4.2.946 Vector

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Vector is a type of Managed_design_object (see 4.2.524). A Vector is a quantity that has magnitude and direction and that is commonly represented by a directed line segment whose length represents the magnitude and whose orientation in space represents the direction.

NOTE EXPRESS specification:

```
*)
ENTITY vector
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- vector

( *
```

4.2.947 Via

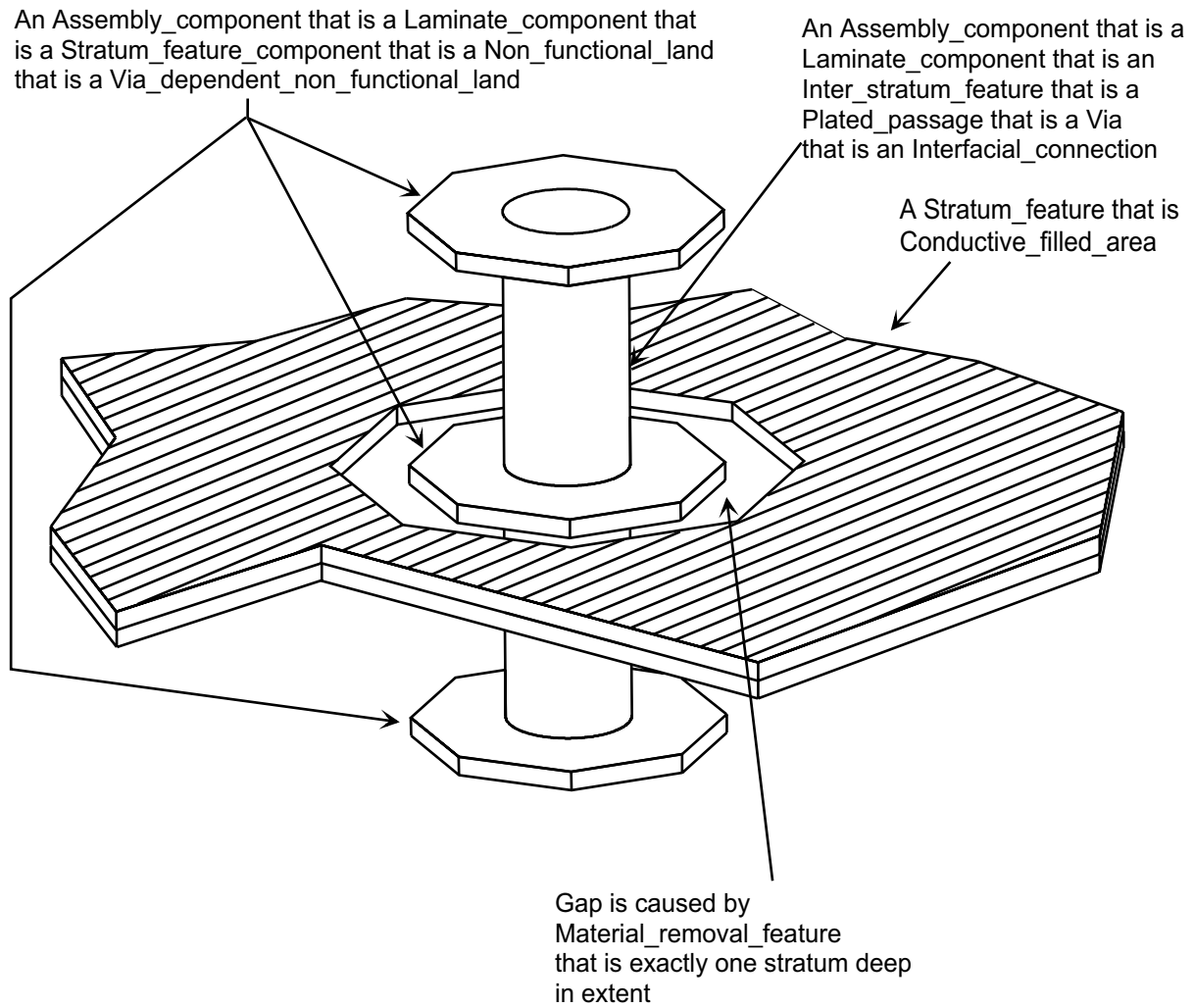
NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Via is a type of Plated_passage (see 4.2.690). A Via is one of a Blind_via (see 4.2.71), a Buried_via (see 4.2.81), or an Interfacial_connection (see 4.2.474). Via is defined in IEC 60050-541.

EXAMPLE 1 Figure 111 illustrates a padstack that contains a Via.

The data associated with a Via are the following:

— usage_definition.

**Figure 111 – Via**

4.2.947.1 usage_definition

The usage_definition specifies the role of the Via_template (see 4.2.952) for the Via.

Constraints:

WR1: A Via shall not interface to an external device. This fact shall be represented by a value of the attribute feature_of_size on the supertype Inter_stratum_feature being set to "FALSE".

NOTE EXPRESS specification:

```
*)
ENTITY via
  ABSTRACT SUPERTYPE OF (ONEOF (blind_via, buried_via, interfacial_connection))
  SUBTYPE OF (plated_passage);
  SELF\assembly_component.usage_definition : via_template;
END_ENTITY; -- via

(*
```

4.2.948 Via_and_contact_size_dependent_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Via_and_contact_size_dependent_land is a type of Land (see 4.2.490) whose shape is dependent upon a Via (see 4.2.947) as well as the size of a Component terminal (see 4.2.137).

The data associated with a Via_and_contact_size_dependent_land are the following:

- interface_access_mechanism;
- reference_via;
- usage_definition.

4.2.948.1 interface_access_mechanism

The interface_access_mechanism is an inverse relationship that specifies that the existence of the Via_and_contact_size_dependent_land is dependent on the existence of the Land_interface_terminal (see 4.2.491) that specifies the Via_and_contact_size_dependent_land as its associated_component. This relationship is for external access only.

4.2.948.2 reference_via

The reference_via specifies the role of the Via (see 4.2.947) for the Via_and_contact_size_dependent_land.

4.2.948.3 usage_definition

The usage_definition specifies the role of the Default_attachment_size_and_via_based_land_physical_template (see 4.2.246) for the Via_and_contact_size_dependent_land.

NOTE EXPRESS specification:

```

*)
ENTITY via_and_contact_size_dependent_land
  SUBTYPE OF (land);
    SELF\assembly_component.usage_definition :
default_attachment_size_and_via_based_land_physical_template;
    reference_via                               : via;
INVERSE
  interface_access_mechanism : land_interface_terminal FOR associated_component;
END_ENTITY; -- via_and_contact_size_dependent_land

( *
```

4.2.949 Via_and_contact_size_dependent_non_functional_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Via_and_contact_size_dependent_non_functional_land is a type of Non_functional_land (see 4.2.562) whose shape is dependent on a Via (see 4.2.947) as well as the size of a Component terminal (see 4.2.137).

NOTE The shape is primarily a function of the mating terminal.

The data associated with a Via_and_contact_size_dependent_non_functional_land are the following:

- interface_access_mechanism;
- reference_via;
- usage_definition.

4.2.949.1 interface_access_mechanism

The `interface_access_mechanism` is an inverse relationship that specifies that the existence of the `Via_and_contact_size_dependent_non_functional_land` is dependent on the existence of the `Non_functional_land_interface_terminal` (see 4.2.563) that specifies the `Via_and_contact_size_dependent_non_functional_land` as its `associated_component`. This relationship is for external access only.

4.2.949.2 reference_via

The `reference_via` specifies the role of the `Via` (see 4.2.947) for the `Via_and_contact_size_dependent_non_functional_land`.

4.2.949.3 usage_definition

The `usage_definition` specifies the role of the `Default_via_based_land_physical_template` (see 4.2.255) for the `Via_and_contact_size_dependent_non_functional_land`.

NOTE EXPRESS specification:

```

*)
ENTITY via_and_contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
    reference_via : via;
    SELF\assembly_component.usage_definition :
default_attachment_size_and_via_based_land_physical_template;
INVERSE
  interface_access_mechanism : non_functional_land_interface_terminal FOR associated_component;
END_ENTITY; -- via_and_contact_size_dependent_non_functional_land

( *
```

4.2.950 Via_dependent_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Via_dependent_land` is a type of `Land` (see 4.2.490) whose shape is dependent only on a `Via` (see 4.2.947).

The data associated with a `Via_dependent_land` are the following:

- `reference_via`;
- `usage_definition`.

4.2.950.1 reference_via

The reference_via specifies the role of the Via (see 4.2.947) for the Via_dependent_land.

4.2.950.2 usage_definition

The usage_definition specifies the role of the Default_via_based_land_physical_template (see 4.2.255) for the Via_dependent_land.

NOTE EXPRESS specification:

```

*)
ENTITY via_dependent_land
  SUBTYPE OF (land);
    reference_via                      : via;
    SELF\assembly_component.usage_definition :
default_via_based_land_physical_template;
END_ENTITY; -- via_dependent_land

( *
```

4.2.951 Via_dependent_non_functional_land

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Via_dependent_non_functional_land is a type of Non_functional_land (see 4.2.562) whose shape is dependent only on a Via (see 4.2.947).

The data associated with a Via_dependent_non_functional_land are the following:

- reference_via;
- usage_definition.

4.2.951.1 reference_via

The reference_via specifies the role of the Via (see 4.2.947) for the Via_dependent_non_functional_land.

4.2.951.2 usage_definition

The usage_definition specifies the role of the Default_via_based_land_physical_template (see 4.2.255) for the Via_dependent_non_functional_land.

NOTE EXPRESS specification:

```

*)
ENTITY via_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
    reference_via : via;
    SELF\assembly_component.usage_definition :
default_via_based_land_physical_template;
END_ENTITY; -- via_dependent_non_functional_land

( *

```

4.2.952 Via_template

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Via_template is a type of Inter_stratum_feature_template (see 4.2.435).

NOTE A Via_template provides a template for a plated passage that supports the realization of connectivity. The access_mechanisms attribute relates the Via_template to two or more terminals, thereby capturing complete connectivity information.

The data associated with a Via_template are the following:

- access_mechanisms;
- of_passage_technology.

4.2.952.1 access_mechanisms

The access_mechanisms is an inverse relationship that specifies that the existence of the Via_template is dependent on the existence of the Via_template_terminal (see 4.2.953) that specifies the Via_template as its associated_definition. There shall be two or more Via_template_terminals for a Via_template.

4.2.952.2 of_passage_technology

The of_passage_technology specifies the role of the Default_via_definition (see 4.2.256) for the Via_template.

NOTE EXPRESS specification:

```

*)
ENTITY via_template
  SUBTYPE OF (inter_stratum_feature_template);
  SELF\inter_stratum_feature_template.of_passage_technology :
  default_via_definition;
  INVERSE
    access_mechanisms : SET [2:?] OF via_template_terminal FOR
    associated_definition;
END_ENTITY; -- via_template

( *
```

4.2.953 Via_template_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Via_template_terminal` is a type of `Shape_element` (see 4.2.810). A `Via_template_terminal` is a terminal in a `Via_template` (see 4.2.952). A `Via_template_terminal` is a terminal that is congruent with the exterior surface of the cylinder of the `Via` (see 4.2.947), including material covered by joined members of `Land` (see 4.2.490). The fabrication join capability is optionally constrained by a set of `Stratum` (see 4.2.845).

The data associated with a `Via_template_terminal` are the following:

- `associated_definition`;
- `disallowed_inter_stratum_extent`;
- `name`;
- `terminal_connection_zone`.

EXAMPLE 1 Figure 112 illustrates an abstraction of the cylinder associated with a `Via` and the associated terminal. Because a `Via` is always an internal design object, there are no interface or join subtypes. A usage view is included that defines the terminal to be just the external wall of the `Via`.

4.2.953.1 associated_definition

The `associated_definition` specifies the role of the `Via_template` (see 4.2.952) for the `Via_template_terminal`.

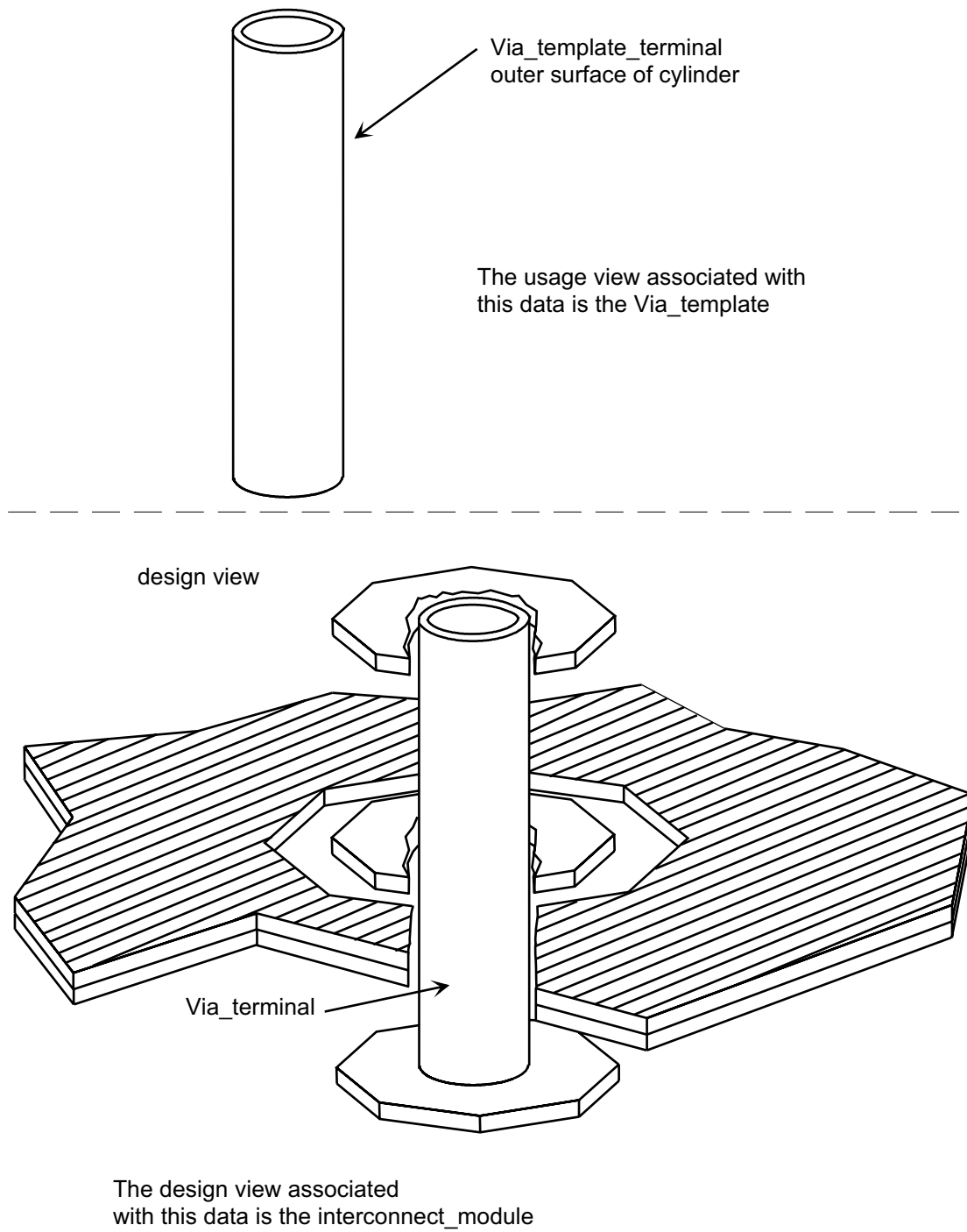


Figure 112 – Via_template_terminal

4.2.953.2 **disallowed_inter_stratum_extent**

The `disallowed_inter_stratum_extent` specifies the `Inter_stratum_extent` (see 4.2.431) for which a connection to the `Via_template_terminal` is disallowed. The `disallowed_inter_stratum_extent` need not be supplied for a `Via_template_terminal`. There may be more than one `Inter_stratum_extent` for a `Via_template_terminal`.

4.2.953.3 **name**

The name identifies the `Via_template_terminal`.

4.2.953.4 **terminal_connection_zone**

The `terminal_connection_zone` specifies the role of the `Connection_zone` (see 4.2.185) for the `Via_template_terminal`. There may be one or more `Connection_zone` for the `Via_template_terminal`. The `terminal_connection_zone` is optional and need not be supplied if the data is unavailable in the source system. If the data is available in the source system, the data shall be provided.

Constraints:

UR1: The combination of the name and `associated_definition` shall be unique within a population of `Via_template_terminal`.

NOTE EXPRESS specification:

```

*)
ENTITY via_template_terminal
  SUBTYPE OF (shape_element);
    associated_definition          : via_template;
    disallowed_inter_stratum_extent : OPTIONAL inter_stratum_extent;
    SELF\shape_element.name       : ee_name;
    terminal_connection_zone       : OPTIONAL SET [1:?] OF connection_zone;
INVERSE
  terminal_shape          : SET [0:1] OF usage_concept_usage_relationship FOR
  associated_usage;
  UNIQUE
    UR1 : name, associated_definition;
END_ENTITY; -- via_template_terminal

( *
```

4.2.954 **Via_terminal**

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Via_terminal` is a type of `Component_terminal` (see 4.2.137). A `Via_terminal` is associated with the larger diameter wall of the `Via` that geometrically approximates a hollow cylinder with some thickness.

NOTE Since the `Via` itself is not allowed to interface to an item external to the interconnect, there is no concept of a join or interface terminal associated with a `Via`. The `Via_terminal` is collocated with and congruent with the entire external wall of the cylinder and may be fastened to as often as necessary.

The data associated with a `Via_terminal` are the following:

- `associated_component`;
- `definition`.

4.2.954.1 associated_component

The `associated_component` specifies the role of the `Via` (see 4.2.947) for the `Via_terminal`.

4.2.954.2 definition

The `definition` specifies the role of the `Via_template_terminal` (see 4.2.953) for the `Via_terminal`.

NOTE EXPRESS specification:

```
*)
ENTITY via_terminal
  SUBTYPE OF (laminar_component_feature);
    SELF\component_feature.definition          : via_template_terminal;
    SELF\component_feature.associated_component : via;
END_ENTITY; -- via_terminal

(*
```

4.2.955 View_level_non_feature_shape_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `View_level_non_feature_shape_definition` is a type of `Non_feature_shape_definition` (see 4.2.559).

The data associated with a `View_level_non_feature_shape_definition` are the following:

- `model_shape`.

4.2.955.1 model_shape

The `model_shape` specifies the role of the `Feature_shape_definition` (see 4.2.372) for the `View_level_non_feature_shape_definition`.

NOTE EXPRESS specification:

```

*)
ENTITY view_level_non_feature_shape_definition
  SUBTYPE OF (non_feature_shape_definition);
  SELF\non_feature_shape_definition.model_shape : view_shape_select;
END_ENTITY; -- view_level_non_feature_shape_definition
( *

```

4.2.956 Viewing_plane

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Viewing_plane is a type of Non_feature_shape_element (see 4.2.560) that is a plane to which a Tolerance_zone_boundary (see 4.2.915) is oriented.

The data associated with a Viewing_plane are the following:

— viewing_plane_specification.

4.2.956.1 viewing_plane_specification

A viewing_plane_specification is an inverse relationship that specifies that the existence of the Viewing_plane is dependent on the existence of the Viewing_plane_definition (see 4.2.958) that specifies the Viewing_plane as its defined_viewing_plane.

NOTE EXPRESS specification:

```

*)
ENTITY viewing_plane
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    viewing_plane_specification : viewing_plane_definition FOR
                                defined_viewing_plane;
END_ENTITY; -- viewing_plane
( *

```

4.2.957 Viewing_plane_based_orientation

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Viewing_plane_based_orientation is a type of Managed_design_object (see 4.2.524). A Viewing_plane_based_orientation is a definition that characterizes the orientation of a Tolerance_zone (see 4.2.913) with respect to a Viewing_plane (see 4.2.956).

The data associated with a Viewing_plane_based_orientation are the following:

- orienting_relationship;
- viewing_plane_specification.

4.2.957.1 orienting_relationship

The orienting_relationship specifies the Tolerance_zone_and_viewing_plane_orienting_relationship (see 4.2.914) that associates the Tolerance_zone (see 4.2.913) with the Viewing_plane (see 4.2.956) with which it is oriented.

4.2.957.2 viewing_plane_specification

The viewing_plane_specification specifies a Viewing_plane_definition (see 4.2.958) that characterizes the Viewing_plane (see 4.2.956) with which the Tolerance_zone (see 4.2.913) is oriented.

Constraints:

WR1: The Viewing_plane_based_orientation shall be specified as the zone_orientation by exactly one Tolerance_zone_definition_with_specified_orientation.

WR2: The Viewing_plane specified as the defined_viewing_plane by the Viewing_plane_definition that is specified as the viewing_plane_specification shall be the same instance of Viewing_plane as specified as the orienting_viewing_plane by the Tolerance_zone_and_viewing_plane_orienting_relationship that is specified as the orienting_relationship.

NOTE EXPRESS specification:

```

*)
ENTITY viewing_plane_based_orientation
  SUBTYPE OF (managed_design_object);
    orienting_relationship      :
tolerance_zone_and_viewing_plane_orienting_relationship;
    viewing_plane_specification : viewing_plane_definition;
END_ENTITY; -- viewing_plane_based_orientation

( *
```

4.2.958 Viewing_plane_definition

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Viewing_plane_definition is a type of Managed_design_object (see 4.2.524). A Viewing_plane_definition is a definition that characterizes a Viewing_plane (see 4.2.956).

The data associated with a `Viewing_plane_definition` are the following:

- `defined_viewing_plane`;
- `viewing_plane_representation`;
- `zone_orientation`.

4.2.958.1 `defined_viewing_plane`

The `defined_viewing_plane` specifies the `Viewing_plane` (see 4.2.956) that is characterized by the `Viewing_plane_definition`, and with which the specified `Tolerance_zone` (see 4.2.913) is oriented.

4.2.958.2 `viewing_plane_representation`

The `viewing_plane_representation` specifies the `Axis_placement` (see 4.2.56) that characterizes the `Viewing_plane` (see 4.2.956) for the `Viewing_plane_definition`.

4.2.958.3 `zone_orientation`

The `zone_orientation` is an inverse relationship that specifies that the existence of the `Viewing_plane_definition` is dependent on the existence of the `Viewing_plane_based_orientation` (see 4.2.957) that specifies the `Viewing_plane_definition` as its `viewing_plane_specification`.

NOTE EXPRESS specification:

```

*)
ENTITY viewing_plane_definition
  SUBTYPE OF (managed_design_object);
    defined_viewing_plane      : viewing_plane;
    viewing_plane_representation : axis_placement;
  INVERSE
    zone_orientation : viewing_plane_based_orientation FOR
                      viewing_plane_specification;
END_ENTITY; -- viewing_plane_definition

--tt tc1 wd9 added
( *
```

4.2.959 `Volume_data_element`

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A `Volume_data_element` is a

The data associated with a Volume_data_element are the following:

— additional_data.

4.2.959.1 additional_data

Constraints:

UR1:

WR1: NOTE EXPRESS specification:

```
*)
ENTITY volume_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- volume_data_element

( *
```

4.2.960 Volume_shape_intersection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Volume_shape_intersection is a type of Managed_design_object (see 4.2.524). A Volume_shape_intersection may be one of Edge_segment_cross_section (see 4.2.311), or Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship (see 4.2.714), or Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship (see 4.2.712), or Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship (see 4.2.710). A Volume_shape_intersection is the projection into two dimensions of a three-dimensional shape based on the intersection of a plane with the three-dimensional shape.

The data associated with a Volume_shape_intersection are the following:

- orientation_intersection_plane;
- planar_shape_origin;
- projected_volume;
- surface_projection.

4.2.960.1 orientation_intersection_plane

The orientation_intersection_plane specifies the role of the Axis_placement_3d (see 4.2.58) for the Volume_shape_intersection. The Z axis of the Axis_placement_3d shall define the normal of the plane.

4.2.960.2 planar_shape_origin

The planar_shape_origin specifies the role of the Axis_placement_2d (see 4.2.57) for the Volume_shape_intersection. The Axis_placement_2d shall define the two dimensional coordinate system.

4.2.960.3 projected_volume

The projected_volume specifies the role of the Bound_volume_shape (see 4.2.76) for the Volume_shape_intersection.

4.2.960.4 surface_projection

The surface_intersection_result specifies the role of the Planar_projected_shape (see 4.2.677) for the Volume_shape_intersection.

NOTE EXPRESS specification:

```

*)
ENTITY volume_shape_intersection
  SUPERTYPE OF (ONEOF(edge_segment_cross_section,
    pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship,
    pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship,
    pre_defined_parallel_datum_axis_symbol_3d_2d_relationship))
  SUBTYPE OF (managed_design_object);
    surface_intersection_result      : planar_shape;
    projected_volume                 : bound_volume_shape;
    orientation_intersection_plane   : axis_placement_3d;
    planar_shape_origin              : axis_placement_2d;
  ( *
WHERE
  WR1: orientation_intersection_plane in projected_volume;
  WR2: planar_shape_origin in surface_intersection_result;
  WR3: The A0 instance referenced by surface_intersection_result shall
        be the A0 instance referenced by projected_volume;
  *)
END_ENTITY; -- volume_shape_intersection

( *
```

4.2.961 Volume_shape_plane_projection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Volume_shape_plane_projection is a type of Managed_design_object (see 4.2.524).

NOTE EXPRESS specification:

```

*)
ENTITY volume_shape_plane_projection
  SUBTYPE OF (managed_design_object);
    surface_projection_result      : planar_shape;
    projected_volume               : bound_volume_shape;
--tt tc1 wd9 modified
    orientation_viewing_plane      : axis_placement_3d;
    viewing_plane                  : axis_placement_3d;
    planar_shape_origin            : axis_placement_2d;
  ( *
WHERE
  WR1: orientation_viewing_plane in projected_volume;
  WR2: planar_shape_origin in surface_projection_result;
  WR3: viewing_plane in projected_volume;
  WR3: The A0 instance referenced by surface_project_result shall
        be the A0 instance referenced by projected_volume;
  *)
END_ENTITY; -- volume_shape_plane_projection

--tt tc1 wd9 modified(change name)
( *

```

4.2.962 Volume_shape_point_projection

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Volume_shape_point_projection is a type of Managed_design_object (see 4.2.524).

NOTE EXPRESS specification:

```

*)
ENTITY volume_shape_point_projection
  SUBTYPE OF (managed_design_object);
    surface_projection_result      : planar_shape;
    projected_volume               : bound_volume_shape;
--tt tc1 wd9 modified
    orientation_viewing_plane      : axis_placement_3d;
    viewing_point                  : cartesian_point;
    planar_shape_origin            : axis_placement_2d;
  ( *
WHERE
  WR1: orientation_viewing_plane in projected_volume;
  WR2: planar_shape_origin in surface_projection_result;
  WR3: viewing_point in projected_volume;
  WR3: The A0 instance referenced by surface_project_result shall
        be the A0 instance referenced by projected_volume;
  *)
END_ENTITY; -- volume_shape_point_projection

( *

```

4.2.963 Wire_terminal

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Wire_terminal is a type of a Package_terminal (see 4.2.581) whose core_material is inherently flexible and ready for installation to a Pca (see 4.2.639) without prior formal preparation by shape modification. The shape of the Wire_terminal is incompletely defined in the library.

NOTE The Feature_shape_occurrence (see ??) inherited by the terminal may be the same Feature_shape_occurrence that is specified by the associated Connection_zone (see 4.2.185). In this case the whole terminal is the Connection_zone.

The data associated with a Wire_terminal are the following:

- internal_connection_zone;
- maximum_wire_terminal_length;
- minimum_wire_terminal_length.

4.2.963.1 internal_connection_zone

The internal_connection_zone specifies the role of a Connection_zone (see 4.2.185) for the Wire_terminal. There shall be one or more Connection_zone for the Wire_terminal. The internal_connection_zone is mandatory because it locates the end of the wire that is the attachment area on the package body.

4.2.963.2 maximum_wire_terminal_length

The maximum_wire_terminal_length specifies the Length_data_element (see 4.2.511) for the maximum length of the Wire_terminal.

4.2.963.3 minimum_wire_terminal_length

The minimum_wire_terminal_length specifies the Length_data_element (see 4.2.511) for the minimum length of the Wire_terminal.

NOTE EXPRESS specification:

```

*)
ENTITY wire_terminal
  SUBTYPE OF (package_terminal);
    SELF\package_terminal.internal_connection_zone : SET [1:?] OF connection_zone;
    maximum_wire_terminal_length                   : length_data_element;
    minimum_wire_terminal_length                   : length_data_element;
END_ENTITY; -- wire_terminal

( *
```

4.2.964 Work_order

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Work_order is a type of Managed_design_object (see 4.2.524). A Work_order is one of a Change_order (see 4.2.96) or a Start_order (see 4.2.838), and is a document that authorizes work in the development of an initial or modified design of a product. A Work_order is the result of the processing of one or more Work_request (see 4.2.965).

The data associated with a Work_order are the following:

- additional_data;
- analysis_data;
- authorizing_agreement;
- initiator;
- referenced_work_request;
- release_date;
- resulting_product;
- start_date;
- work_order_approval;
- work_order_id.

4.2.964.1 additional_data

The additional_data specifies the Ee_text (see 4.2.339) that provides additional pertinent information that was compiled during the development of the related set of Work_request (see 4.2.965) for the Work_order. The additional_data need not be specified for a particular Work_order.

4.2.964.2 analysis_data

The analysis_data specifies the Ee_text (see 4.2.339) for the result of an assessment performed to validate the design or modification in a related set of Work_requests for the Work_order. The analysis_data need not be specified for a particular Work_order.

4.2.964.3 authorizing_agreement

The authorizing_agreement specifies the role of the Contract (see 4.2.196) for the Work_order. The authorizing_agreement need not be provided for a Work_order.

4.2.964.4 initiator

The initiator specifies either the Person_and_organization (see 4.2.647) or the Organization (see 4.2.569), that initiates the Work_order. There shall be one or more Person_and_organization or Organization for the Work_order.

4.2.964.5 referenced_work_request

The referenced_work_request specifies the Organization (see 4.2.569) that initiated this Work_order. There shall be one or more Organization (see 4.2.569) for a Work_order.

4.2.964.6 release_date

The release_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) upon which the Work_order is released.

4.2.964.7 resulting_product

The resulting_product specifies the Ee_product_version (see 4.2.328) of the objects that are the subject of the Work_order. There shall be one or more Ee_product_version for a Work_order.

4.2.964.8 start_date

The start_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) upon which the Work_order is begun.

4.2.964.9 work_order_approval

The work_order_approval specifies the Ee_approval (see 4.2.314) that applies to the Work_order.

4.2.964.10 work_order_id

The work_order_id specifies the identifier of the Work_order.

Constraints:

UR1: The work_order_id shall be unique within a population of Work_order.

WR1: The resulting_product and the referenced_work_request shall specify the same Ee_product_version.

NOTE EXPRESS specification:

```

*)
ENTITY work_order
  ABSTRACT SUPERTYPE OF (ONEOF (change_order,start_order))
  SUBTYPE OF (managed_design_object);
    referenced_work_request : SET [1:?] OF work_request;
    work_order_id           : ee_name;
    resulting_product       : SET [1:?] OF ee_product_version;
    analysis_data           : OPTIONAL ee_text;
    additional_data         : OPTIONAL ee_text;
    work_order_approval     : ee_approval;
    initiator               : SET [1:?] OF person_organization;
    start_date              : date_or_date_and_time;
    release_date            : date_or_date_and_time;
    authorizing_agreement   : OPTIONAL contract;
  UNIQUE
    UR1 : work_order_id;
END_ENTITY; -- work_order

( *
```

4.2.965 Work_request

NOTE The text description for this Application Object has not been fully updated to agree with the EXPRESS.

A Work_request is a type of Managed_design_object (see 4.2.524). A Work_request is either a Change_request (see 4.2.97) or a Start_request (see 4.2.839), a document that identifies the work to be done to initiate or redefine the design of a part.

The data associated with a Work_request are the following:

- description;
- implementation_order;
- initiator;
- reason;
- referenced_product;
- request_date;
- request_recipient;
- status;
- work_request_approval;
- work_request_id.

4.2.965.1 description

The description specifies the Ee_text (see 4.2.339) information of the work to be done to create or change the design of a product by a Work_request.

4.2.965.2 implementation_order

The implementation_order is an inverse relationship that specifies the Work_order (see 4.2.964) that specifies the Work_request as its referenced_work_request. There shall be no more than one Work_order for a Work_request.

4.2.965.3 initiator

The initiator specifies either the Person_and_organization (see 4.2.647) or the Organization that initiates the Work_request. There shall be one or more Person_and_organization or Organization (see 4.2.569) for a Work_request.

4.2.965.4 reason

The reason specifies the Ee_text (see 4.2.339) or the Evaluated_characteristic (see 4.2.356) which is the purpose for generating the Work_request.

4.2.965.5 referenced_product

The referenced_product specifies the specific Ee_product_version (see 4.2.328) that shall be the subject of the Work_request. There shall be one or more Ee_product_version for the Work_request.

4.2.965.6 request_date

The request_date specifies either the Date (see 4.2.212) or the Date_and_time (see 4.2.213) when the Work_request was initiated.

4.2.965.7 request_recipient

The request_recipient specifies either the Person_and_organization (see 4.2.647) or the Organization (see 4.2.569) that will receive the Work_request and take whatever action is called out in it. There shall be one or more Person_and_organization or Organization for a Work_request.

4.2.965.8 status

The status specifies the text that describes the current level of completion of the Work_request.

EXAMPLE 1 Example values for the status of a Work_request may be proposed, in-work, issued, and hold.

4.2.965.9 work_request_approval

The work_request_approval specifies the Ee_approval (see 4.2.314) for the Work_request before it is to be acted upon.

4.2.965.10 work_request_id

The work_request_id specifies the Ee_text (see 4.2.339) identifier for the proposed work on a product's design.

Constraints:

UR1: The work_request_id shall be unique in a population of Work_request.

IP1: The work_request_id shall be unique in a population of Work_request in an Organization.

IP2: Each member of Work_request shall have at least one member of Change_request or Start_request in the role of change_or_start_request.

NOTE EXPRESS specification:

```

*)
ENTITY work_request
  ABSTRACT SUPERTYPE OF (ONEOF (start_request,change_request))
  SUBTYPE OF (managed_design_object);
  work_request_approval : ee_approval;
  request_recipient      : SET [1:?] OF person_organization;
  work_request_id        : ee_name;
  referenced_product     : SET [1:?] OF ee_product_version;
  request_date           : date_or_date_and_time;
  description            : ee_text;
  reason                 : evaluated_characteristic_or_description;
  initiator              : SET [1:?] OF person_organization;
  status                 : STRING;
  INVERSE
    implementation_order : SET [0:1] OF work_order FOR referenced_work_request;
  UNIQUE
    UR1 : work_request_id;
END_ENTITY; -- work_request
( *

```

4.3 Application assertions

This subclause specifies the application assertions for the application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

4.3.1 Add_design_object_management_relationship to Managed_design_object

Each Add_design_object_management_relationship has current_design_object defined by exactly one Managed_design_object. Each Managed_design_object defines current_design_object for zero, one, or more Add_design_object_management_relationship objects.

4.3.2 Adjacent_stratum_surface_definition to Stratum_surface

Each Adjacent_stratum_surface_definition has subsequent_surface defined by exactly one Stratum_surface. Each Stratum_surface defines subsequent_surface for zero, one, or more Adjacent_stratum_surface_definition objects. Each Adjacent_stratum_surface_definition has precedent_surface defined by exactly one Stratum_surface. Each Stratum_surface defines precedent_surface for zero, one, or more Adjacent_stratum_surface_definition objects.

4.3.3 Adjacent_stratum_surface_embedded_component_surface_definition to Stratum_surface

Each Adjacent_stratum_surface_embedded_component_surface_definition has subsequent_item defined by zero or one Stratum_surface objects. Each Stratum_surface defines subsequent_item for zero, one, or more Adjacent_stratum_surface_embedded_component_surface_definition objects. This assertion is established through a SELECT type. Each Adjacent_stratum_surface_embedded_component_surface_definition has precedent_item defined by zero or one Stratum_surface objects. Each Stratum_surface defines precedent_item for zero, one, or more Adjacent_stratum_surface_embedded_component_surface_definition objects. This assertion is established through a SELECT type.

4.3.4 Adjacent_stratum_surface_embedded_component_surface_definition to Component_feature

Each Adjacent_stratum_surface_embedded_component_surface_definition has subsequent_item defined by zero or one Component_feature objects. Each Component_feature defines subsequent_item for zero, one, or more Adjacent_stratum_surface_embedded_component_surface_definition objects. This assertion is established through a SELECT type. Each Adjacent_stratum_surface_embedded_component_surface_definition has precedent_item defined by zero or one Component_feature objects. Each Component_feature defines precedent_item for zero, one, or more Adjacent_stratum_surface_embedded_component_surface_definition objects. This assertion is established through a SELECT type.

4.3.5 Aggregate_connectivity_requirement to Design_composition_path

Each Aggregate_connectivity_requirement has design_definition_path defined by zero or one Design_composition_path objects. Each Design_composition_path defines design_definition_path for zero, one, or more Aggregate_connectivity_requirement objects. This assertion is established through a SELECT type.

4.3.6 Aggregate_connectivity_requirement to Functional_unit_network_definition

Each Aggregate_connectivity_requirement has design_definition_path defined by zero or one Functional_unit_network_definition objects. Each Functional_unit_network_definition defines design_definition_path for zero, one, or more Aggregate_connectivity_requirement objects. This assertion is established through a SELECT type.

4.3.7 Aggregate_connectivity_requirement to Functional_unit_network_node_definition

Each Aggregate_connectivity_requirement has connectivity_requirement defined by exactly one Functional_unit_network_node_definition. Each Functional_unit_network_node_definition defines connectivity_requirement for zero, one, or more Aggregate_connectivity_requirement objects.

4.3.8 Altered_package to Package

Each Altered_package has package_to_be_altered defined by exactly one Package. Each Package defines package_to_be_altered for zero, one, or more Altered_package objects.

4.3.9 Altered_packaged_part to Packaged_part

Each Altered_packaged_part has base_packaged_part defined by exactly one Packaged_part. Each Packaged_part defines base_packaged_part for zero, one, or more Altered_packaged_part objects.

4.3.10 Altered_packaged_part to Altered_package

Each Altered_packaged_part has used_package defined by exactly one Altered_package. Each Altered_package defines used_package for zero, one, or more Altered_packaged_part objects.

4.3.11 Altered_package_terminal to Altered_package

Each Altered_package_terminal has associated_definition defined by exactly one Altered_package. Each Altered_package defines associated_definition for zero, one, or more Altered_package_terminal objects.

4.3.12 Altered_package_terminal to Package_terminal

Each Altered_package_terminal has terminal_to_be_altered defined by exactly one Package_terminal. Each Package_terminal defines terminal_to_be_altered for zero, one, or more Altered_package_terminal objects.

4.3.13 Alternate_product to Ee_product

Each Alternate_product has defined_alternate_product defined by exactly one Ee_product. Each Ee_product defines defined_alternate_product for zero, one, or more Alternate_product objects.

4.3.14 Alternate_select_product to Contract

Each Alternate_select_product has authorizing_agreement defined by exactly one Contract. Each Contract defines authorizing_agreement for zero, one, or more Alternate_select_product objects.

4.3.15 Alternate_select_product to Ee_product

Each Alternate_select_product has defining_base_product defined by exactly one Ee_product. Each Ee_product defines defining_base_product for zero, one, or more Alternate_select_product objects. Each Alternate_select_product has defined_alternate_product defined by one or more Ee_product objects. Each Ee_product defines defined_alternate_product for zero, one, or more Alternate_select_product objects.

4.3.16 Analytical_model to Analytical_model_definition

Each Analytical_model has associated_definition defined by exactly one Analytical_model_definition. Each Analytical_model_definition defines associated_definition for zero, one, or more Analytical_model objects.

4.3.17 Analytical_model to Ee_specification

Each Analytical_model has source_code defined by exactly one Ee_specification. Each Ee_specification defines source_code for zero, one, or more Analytical_model objects.

4.3.18 Analytical_model to Model_parameter

Each Analytical_model has parameter defined by one or more Model_parameter objects. Each Model_parameter defines parameter for zero, one, or more Analytical_model objects.

4.3.19 Analytical_model to Ee_document

Each Analytical_model has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Analytical_model objects.

4.3.20 Analytical_model to Language_reference_manual

Each Analytical_model has representation_language defined by exactly one Language_reference_manual. Each Language_reference_manual defines representation_language for zero, one, or more Analytical_model objects.

4.3.21 Analytical_model_make_from_relationship to Analytical_model

Each Analytical_model_make_from_relationship has resultant_model defined by exactly one Analytical_model. Each Analytical_model defines resultant_model for zero, one, or more Analytical_model_make_from_relationship objects. Each Analytical_model_make_from_relationship has reusable_model defined by exactly one Analytical_model. Each Analytical_model defines reusable_model for zero, one, or more Analytical_model_make_from_relationship objects.

4.3.22 Analytical_model_port to Analytical_model

Each Analytical_model_port has accessed_analytical_model defined by exactly one Analytical_model. Each Analytical_model defines accessed_analytical_model for one or more Analytical_model_port objects.

4.3.23 Analytical_model_port_assignment to Analytical_model_port

Each Analytical_model_port_assignment has assigned_analytical_model_port defined by exactly one Analytical_model_port. Each Analytical_model_port defines assigned_analytical_model_port for zero, one, or more Analytical_model_port_assignment objects.

4.3.24 Analytical_model_port_assignment to Functional_unit_usage_view_terminal_definition

Each Analytical_model_port_assignment has assigned_functional_unit_terminal defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines assigned_functional_unit_terminal for zero, one, or more Analytical_model_port_assignment objects.

4.3.25 Analytical_model_port_make_from_relationship to Analytical_model_port

Each Analytical_model_port_make_from_relationship has resultant_feature defined by exactly one Analytical_model_port. Each Analytical_model_port defines resultant_feature for zero, one, or more Analytical_model_port_make_from_relationship objects. Each Analytical_model_port_make_from_relationship has reusable_feature defined by exactly one Analytical_model_port. Each Analytical_model_port defines reusable_feature for zero, one, or more Analytical_model_port_make_from_relationship objects.

4.3.26 Analytical_model_port_make_from_relationship to Analytical_model_make_from_relationship

Each Analytical_model_port_make_from_relationship has associated_make_from defined by exactly one Analytical_model_make_from_relationship. Each Analytical_model_make_from_relationship defines associated_make_from for zero, one, or more Analytical_model_port_make_from_relationship objects.

4.3.27 Analytical_representation to Analytical_model

Each Analytical_representation has model_representation defined by exactly one Analytical_model. Each Analytical_model defines model_representation for zero, one, or more Analytical_representation objects.

4.3.28 Analytical_representation to Parameter_assignment

Each Analytical_representation has model_parameter_assignment defined by one or more Parameter_assignment objects. Each Parameter_assignment defines model_parameter_assignment for zero, one, or more Analytical_representation objects.

4.3.29 Angular_dimension_with_direction_vector to Axis_placement

Each Angular_dimension_with_direction_vector has measurement_orientation defined by zero or one Axis_placement objects. Each Axis_placement defines measurement_orientation for zero, one, or more Angular_dimension_with_direction_vector objects. This assertion is established through a SELECT type.

4.3.30 Angular_dimension_with_direction_vector to Datum_based_vector_orientation

Each Angular_dimension_with_direction_vector has measurement_orientation defined by zero or one Datum_based_vector_orientation objects. Each Datum_based_vector_orientation defines measurement_orientation for zero, one, or more Angular_dimension_with_direction_vector objects. This assertion is established through a SELECT type.

4.3.31 Assembled_with_bonding to Assembly_bond_definition

Each Assembled_with_bonding has default_bond_definition defined by exactly one Assembly_bond_definition. Each Assembly_bond_definition defines default_bond_definition for zero, one, or more Assembled_with_bonding objects.

4.3.32 Assembled_with_fasteners to Assembly_group_component

Each Assembled_with_fasteners has associated_assembly defined by exactly one Assembly_group_component. Each Assembly_group_component defines associated_assembly for zero, one, or more Assembled_with_fasteners objects.

4.3.33 Assembly_alternate_product to Physical_unit_design_view

Each Assembly_alternate_product has basis_context defined by exactly one Physical_unit_design_view. Each Physical_unit_design_view defines basis_context for zero, one, or more Assembly_alternate_product objects.

4.3.34 Assembly_bond_definition to Part_feature

Each Assembly_bond_definition has bonded_feature_2 defined by exactly one Part_feature. Each Part_feature defines bonded_feature_2 for zero, one, or more Assembly_bond_definition objects. Each Assembly_bond_definition has bonded_feature_1 defined by exactly one Part_feature. Each Part_feature defines bonded_feature_1 for zero, one, or more Assembly_bond_definition objects.

4.3.35 Assembly_bond_definition to Ee_material

Each Assembly_bond_definition has bond_material defined by exactly one Ee_material. Each Ee_material defines bond_material for zero, one, or more Assembly_bond_definition objects.

4.3.36 Assembly_bond_definition to External_definition

Each Assembly_bond_definition has bond_category defined by exactly one External_definition. Each External_definition defines bond_category for zero, one, or more Assembly_bond_definition objects.

4.3.37 Assembly_bond_definition to Externally_defined_ee_product_definition

Each Assembly_bond_definition has associated_definition defined by exactly one Externally_defined_ee_product_definition. Each Externally_defined_ee_product_definition defines associated_definition for zero, one, or more Assembly_bond_definition objects.

4.3.38 Assembly_bond_definition to Analytical_representation

Each Assembly_bond_definition has analytical_assembly_bond_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines analytical_assembly_bond_representation for zero, one, or more Assembly_bond_definition objects.

4.3.39 Assembly_component to Part_template

Each Assembly_component has usage_definition defined by zero or one Part_template objects. Each Part_template defines usage_definition for zero, one, or more Assembly_component objects. This assertion is established through a SELECT type.

4.3.40 Assembly_component to Physical_unit

Each Assembly_component has usage_definition defined by zero or one Physical_unit objects. Each Physical_unit defines usage_definition for zero, one, or more Assembly_component objects. This assertion is established through a SELECT type.

4.3.41 Assembly_component_2d_shape to Assembly_component

Each Assembly_component_2d_shape has shape_characterized_component defined by one or more Assembly_component objects. Each Assembly_component defines shape_characterized_component for zero or one Assembly_component_2d_shape objects.

4.3.42 Assembly_component_3d_shape to Assembly_component

Each Assembly_component_3d_shape has shape_characterized_component defined by one or more Assembly_component objects. Each Assembly_component defines shape_characterized_component for zero or one Assembly_component_3d_shape objects.

4.3.43 Assembly_component_non_planar_shape to Assembly_component

Each Assembly_component_non_planar_shape has shape_characterized_component defined by one or more Assembly_component objects. Each Assembly_component defines shape_characterized_component for zero or one Assembly_component_non_planar_shape objects.

4.3.44 Assembly_component_placement_link to Assembly_component

Each Assembly_component_placement_link has subsequent_element defined by exactly one Assembly_component. Each Assembly_component defines subsequent_element for zero, one, or more Assembly_component_placement_link objects. Each Assembly_component_placement_link has precedent_element defined by exactly one Assembly_component. Each Assembly_component defines precedent_element for zero, one, or more Assembly_component_placement_link objects.

4.3.45 Assembly_composition_relationship to Ee_material

Each Assembly_composition_relationship has assembly_bonding_agent defined by one or more Ee_material objects. Each Ee_material defines assembly_bonding_agent for zero, one, or more Assembly_composition_relationship objects.

4.3.46 Assembly_composition_relationship to Security_classification

Each Assembly_composition_relationship has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Assembly_composition_relationship objects.

4.3.47 Assembly_composition_relationship to Assembly_component

Each Assembly_composition_relationship has component defined by exactly one Assembly_component. Each Assembly_component defines component for one or more Assembly_composition_relationship objects.

4.3.48 Assembly_composition_relationship to Physical_unit_design_view

Each Assembly_composition_relationship has assembly defined by exactly one Physical_unit_design_view. Each Physical_unit_design_view defines assembly for zero, one, or more Assembly_composition_relationship objects.

4.3.49 Assembly_joint to Assembly_module

Each Assembly_joint has associated_assembly defined by zero or one Assembly_module objects. Each Assembly_module defines associated_assembly for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.50 Assembly_joint to Assembly_group_component

Each Assembly_joint has associated_assembly defined by zero or one Assembly_group_component objects. Each Assembly_group_component defines associated_assembly for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.51 Assembly_joint to Interconnect_component_join_terminal

Each Assembly_joint has assembly_features_2 defined by zero or one Interconnect_component_join_terminal objects. Each Interconnect_component_join_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Interconnect_component_join_terminal objects. Each Interconnect_component_join_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.52 Assembly_joint to Interconnect_component_interface_terminal

Each Assembly_joint has assembly_features_2 defined by zero or one Interconnect_component_interface_terminal objects. Each Interconnect_component_interface_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Interconnect_component_interface_terminal objects. Each Interconnect_component_interface_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.53 Assembly_joint to Packaged_connector_component_interface_terminal

Each Assembly_joint has assembly_features_2 defined by zero or one Packaged_connector_component_interface_terminal objects. Each Packaged_connector_component_interface_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Packaged_connector_component_interface_terminal objects. Each Packaged_connector_component_interface_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.54 Assembly_joint to Package_terminal_occurrence

Each Assembly_joint has assembly_features_2 defined by zero or one Package_terminal_occurrence objects. Each Package_terminal_occurrence defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Package_terminal_occurrence objects. Each Package_terminal_occurrence defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.55 Assembly_joint to Bare_die_component_terminal

Each Assembly_joint has assembly_features_2 defined by zero or one Bare_die_component_terminal objects. Each Bare_die_component_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Bare_die_component_terminal objects. Each Bare_die_component_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.56 **Assembly_joint to Packaged_component_join_terminal**

Each Assembly_joint has assembly_features_2 defined by zero or one Packaged_component_join_terminal objects. Each Packaged_component_join_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Packaged_component_join_terminal objects. Each Packaged_component_join_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.57 **Assembly_joint to Assembly_module_component_terminal**

Each Assembly_joint has assembly_features_2 defined by zero or one Assembly_module_component_terminal objects. Each Assembly_module_component_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Assembly_module_component_terminal objects. Each Assembly_module_component_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.58 **Assembly_joint to Interconnect_module_component_surface_feature**

Each Assembly_joint has assembly_features_2 defined by zero or one Interconnect_module_component_surface_feature objects. Each Interconnect_module_component_surface_feature defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Interconnect_module_component_surface_feature objects. Each Interconnect_module_component_surface_feature defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.59 **Assembly_joint to Interconnect_module_component_terminal**

Each Assembly_joint has assembly_features_2 defined by zero or one Interconnect_module_component_terminal objects. Each Interconnect_module_component_terminal defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Interconnect_module_component_terminal objects. Each Interconnect_module_component_terminal defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.60 **Assembly_joint to Component_mounting_feature**

Each Assembly_joint has assembly_features_2 defined by zero or one Component_mounting_feature objects. Each Component_mounting_feature defines assembly_features_2 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type. Each Assembly_joint has assembly_features_1 defined by zero or one Component_mounting_feature objects. Each Component_mounting_feature defines assembly_features_1 for zero, one, or more Assembly_joint objects. This assertion is established through a SELECT type.

4.3.61 Assembly_material_composition_relationship to Security_classification

Each Assembly_material_composition_relationship has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Assembly_material_composition_relationship objects.

4.3.62 Assembly_material_composition_relationship to Ee_material

Each Assembly_material_composition_relationship has component_material defined by exactly one Ee_material. Each Ee_material defines component_material for zero, one, or more Assembly_material_composition_relationship objects.

4.3.63 Assembly_material_composition_relationship to Physical_unit_design_view

Each Assembly_material_composition_relationship has assembly defined by exactly one Physical_unit_design_view. Each Physical_unit_design_view defines assembly for zero, one, or more Assembly_material_composition_relationship objects.

4.3.64 Assembly_material_composition_relationship to Data_element

Each Assembly_material_composition_relationship has quantity defined by exactly one Data_element. Each Data_element defines quantity for zero, one, or more Assembly_material_composition_relationship objects.

4.3.65 Assembly_module_component to Assembly_module_usage_view

Each Assembly_module_component has usage_definition defined by exactly one Assembly_module_usage_view. Each Assembly_module_usage_view defines usage_definition for zero, one, or more Assembly_module_component objects.

4.3.66 Assembly_module_component_terminal to Assembly_module_component

Each Assembly_module_component_terminal has associated_component defined by exactly one Assembly_module_component. Each Assembly_module_component defines associated_component for two or more Assembly_module_component_terminal objects.

4.3.67 Assembly_module_component_terminal to Assembly_module_terminal

Each Assembly_module_component_terminal has definition defined by exactly one Assembly_module_terminal. Each Assembly_module_terminal defines definition for zero, one, or more Assembly_module_component_terminal objects.

4.3.68 Assembly_module_macro_component to Assembly_module

Each Assembly_module_macro_component has design_definition defined by exactly one Assembly_module. Each Assembly_module defines design_definition for zero, one, or more Assembly_module_macro_component objects.

4.3.69 Assembly_module_macro_component_join_terminal to Assembly_module_macro_component

Each Assembly_module_macro_component_join_terminal has associated_component defined by exactly one Assembly_module_macro_component. Each Assembly_module_macro_component defines associated_component for one or more Assembly_module_macro_component_join_terminal objects.

4.3.70 Assembly_module_macro_component_join_terminal to Assembly_module_join_terminal

Each Assembly_module_macro_component_join_terminal has definition defined by exactly one Assembly_module_join_terminal. Each Assembly_module_join_terminal defines definition for zero, one, or more Assembly_module_macro_component_join_terminal objects.

4.3.71 Assembly_module_terminal to Connection_zone

Each Assembly_module_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Assembly_module_terminal objects.

4.3.72 Assembly_module_terminal to Assembly_module_usage_view_connector_relationship

Each Assembly_module_terminal has related_connector defined by exactly one Assembly_module_usage_view_connector_relationship. Each Assembly_module_usage_view_connector_relationship defines related_connector for zero, one, or more Assembly_module_terminal objects.

4.3.73 Assembly_module_terminal to Packaged_part_interface_terminal

Each Assembly_module_terminal has reference_terminal defined by exactly one Packaged_part_interface_terminal. Each Packaged_part_interface_terminal defines reference_terminal for zero, one, or more Assembly_module_terminal objects.

4.3.74 Assembly_module_terminal to Assembly_module_usage_view

Each Assembly_module_terminal has associated_definition defined by exactly one Assembly_module_usage_view. Each Assembly_module_usage_view defines associated_definition for two or more Assembly_module_terminal objects.

4.3.75 Assembly_module_usage_view to Datum_based_length_measure

Each Assembly_module_usage_view has maximum_positive_component_height defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines maximum_positive_component_height for zero, one, or more Assembly_module_usage_view objects. Each Assembly_module_usage_view has maximum_negative_component_height defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines maximum_negative_component_height for zero, one, or more Assembly_module_usage_view objects.

4.3.76 Assembly_module_usage_view to Functional_unit_usage_view

Each Assembly_module_usage_view has implemented_function defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines implemented_function for zero, one, or more Assembly_module_usage_view objects.

4.3.77 Assembly_module_usage_view_connector_relationship to Packaged_connector

Each Assembly_module_usage_view_connector_relationship has associated_usage defined by exactly one Packaged_connector. Each Packaged_connector defines associated_usage for zero, one, or more Assembly_module_usage_view_connector_relationship objects.

4.3.78 Assembly_module_usage_view_connector_relationship to Assembly_module_usage_view

Each Assembly_module_usage_view_connector_relationship has associating_usage defined by exactly one Assembly_module_usage_view. Each Assembly_module_usage_view defines associating_usage for zero, one, or more Assembly_module_usage_view_connector_relationship objects.

4.3.79 Assembly_requirement_allocation to Ee_requirement_occurrence

Each Assembly_requirement_allocation has requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines requirement for zero, one, or more Assembly_requirement_allocation objects.

4.3.80 Assembly_requirement_allocation to Assembly_component

Each Assembly_requirement_allocation has assembly_assignment defined by zero or one Assembly_component objects. Each Assembly_component defines assembly_assignment for zero, one, or more Assembly_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.81 Assembly_requirement_allocation to Assembly_joint

Each Assembly_requirement_allocation has assembly_assignment defined by zero or one Assembly_joint objects. Each Assembly_joint defines assembly_assignment for zero, one, or more Assembly_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.82 Assembly_requirement_allocation to Component_feature

Each Assembly_requirement_allocation has assembly_assignment defined by zero or one Component_feature objects. Each Component_feature defines assembly_assignment for zero, one, or more Assembly_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.83 Assembly_separation_requirement_allocation to Assembly_component

Each Assembly_separation_requirement_allocation has disjoint_assignment defined by zero or one Assembly_component objects. Each Assembly_component defines disjoint_assignment for zero, one, or more Assembly_separation_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.84 Assembly_separation_requirement_allocation to Assembly_joint

Each Assembly_separation_requirement_allocation has disjoint_assignment defined by zero or one Assembly_joint objects. Each Assembly_joint defines disjoint_assignment for zero, one, or more Assembly_separation_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.85 Assembly_separation_requirement_allocation to Component_feature

Each Assembly_separation_requirement_allocation has disjoint_assignment defined by zero or one Component_feature objects. Each Component_feature defines disjoint_assignment for zero, one, or more Assembly_separation_requirement_allocation objects. This assertion is established through a SELECT type.

4.3.86 Axis_placement to Cartesian_point

Each Axis_placement has translation defined by exactly one Cartesian_point. Each Cartesian_point defines translation for zero, one, or more Axis_placement objects.

4.3.87 Axis_placement to Orientation

Each Axis_placement has device_orientation defined by exactly one Orientation. Each Orientation defines device_orientation for zero, one, or more Axis_placement objects.

4.3.88 Bare_die to Functional_unit_usage_view

Each Bare_die has implemented_function defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines implemented_function for zero, one, or more Bare_die objects.

4.3.89 Bare_die to Technology

Each Bare_die has unit_technology defined by exactly one Technology. Each Technology defines unit_technology for zero, one, or more Bare_die objects.

4.3.90 Bare_die to Seating_plane

Each Bare_die has die_seating_plane defined by exactly one Seating_plane. Each Seating_plane defines die_seating_plane for zero, one, or more Bare_die objects.

4.3.91 Bare_die to Cartesian_point

Each Bare_die has maximum_material_condition_centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines maximum_material_condition_centroid_location for zero, one, or more Bare_die objects. Each Bare_die has least_material_condition_centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines least_material_condition_centroid_location for zero, one, or more Bare_die objects.

4.3.92 Bare_die to Length_data_element

Each Bare_die has maximum_height_above_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_height_above_seating_plane for zero, one, or more Bare_die objects.

4.3.93 Bare_die_bottom_surface to Bare_die

Each Bare_die_bottom_surface has associated_definition defined by exactly one Bare_die. Each Bare_die defines associated_definition for zero, one, or more Bare_die_bottom_surface objects.

4.3.94 Bare_die_component to Bare_die

Each Bare_die_component has usage_definition defined by exactly one Bare_die. Each Bare_die defines usage_definition for zero, one, or more Bare_die_component objects.

4.3.95 Bare_die_component_terminal to Bare_die_component

Each Bare_die_component_terminal has associated_component defined by exactly one Bare_die_component. Each Bare_die_component defines associated_component for zero, one, or more Bare_die_component_terminal objects.

4.3.96 Bare_die_component_terminal to Bare_die_terminal

Each Bare_die_component_terminal has definition defined by exactly one Bare_die_terminal. Each Bare_die_terminal defines definition for zero, one, or more Bare_die_component_terminal objects.

4.3.97 Bare_die_edge_segment_surface to Bare_die

Each Bare_die_edge_segment_surface has associated_definition defined by exactly one Bare_die. Each Bare_die defines associated_definition for zero, one, or more Bare_die_edge_segment_surface objects.

4.3.98 Bare_die_edge_segment_surface to Edge_segment_vertex

Each Bare_die_edge_segment_surface has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Bare_die_edge_segment_surface objects. Each Bare_die_edge_segment_surface has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Bare_die_edge_segment_surface objects.

4.3.99 Bare_die_edge_segment_surface to Bare_die_edge_surface

Each Bare_die_edge_segment_surface has composed_surface defined by exactly one Bare_die_edge_surface. Each Bare_die_edge_surface defines composed_surface for zero, one, or more Bare_die_edge_segment_surface objects.

4.3.100 Bare_die_edge_surface to Bare_die

Each Bare_die_edge_surface has associated_definition defined by exactly one Bare_die. Each Bare_die defines associated_definition for zero, one, or more Bare_die_edge_surface objects.

4.3.101 Bare_die_terminal to Connection_zone_interface_plane_relationship

Each Bare_die_terminal has seating_plane_zone defined by exactly one Connection_zone_interface_plane_relationship. Each Connection_zone_interface_plane_relationship defines seating_plane_zone for zero or one Bare_die_terminal objects.

4.3.102 Bare_die_terminal to Connection_zone

Each Bare_die_terminal has external_connection_area defined by one or more Connection_zone objects. Each Connection_zone defines external_connection_area for zero, one, or more Bare_die_terminal objects.

4.3.103 Bare_die_terminal to Ee_material

Each Bare_die_terminal has material defined by exactly one Ee_material. Each Ee_material defines material for zero, one, or more Bare_die_terminal objects.

4.3.104 Bare_die_terminal_surface_constituent_relationship to Bare_die_terminal

Each Bare_die_terminal_surface_constituent_relationship has related_shape_element defined by exactly one Bare_die_terminal. Each Bare_die_terminal defines related_shape_element for zero, one, or more Bare_die_terminal_surface_constituent_relationship objects.

4.3.105 Bare_die_terminal_surface_constituent_relationship to Bare_die_surface

Each Bare_die_terminal_surface_constituent_relationship has relating_shape_element defined by exactly one Bare_die_surface. Each Bare_die_surface defines relating_shape_element for zero, one, or more Bare_die_terminal_surface_constituent_relationship objects.

4.3.106 Bare_die_top_surface to Bare_die

Each Bare_die_top_surface has associated_definition defined by exactly one Bare_die. Each Bare_die defines associated_definition for zero, one, or more Bare_die_top_surface objects.

4.3.107 Bevel_edge_feature_shape to Angle_data_element

Each Bevel_edge_feature_shape has maximum_bevel_angle defined by exactly one Angle_data_element. Each Angle_data_element defines maximum_bevel_angle for zero, one, or more Bevel_edge_feature_shape objects. Each Bevel_edge_feature_shape has minimum_bevel_angle defined by exactly one Angle_data_element. Each Angle_data_element defines minimum_bevel_angle for zero, one, or more Bevel_edge_feature_shape objects.

4.3.108 Bevel_edge_feature_shape to Length_data_element

Each Bevel_edge_feature_shape has maximum_bevel_width defined by exactly one Length_data_element. Each Length_data_element defines maximum_bevel_width for zero, one, or more Bevel_edge_feature_shape objects. Each Bevel_edge_feature_shape has minimum_bevel_width defined by exactly one Length_data_element. Each Length_data_element defines minimum_bevel_width for zero, one, or more Bevel_edge_feature_shape objects.

4.3.109 Blind_passage_template to Shape_dimension_representation

Each Blind_passage_template has bottom_distance_from_datum_reference defined by exactly one Shape_dimension_representation. Each Shape_dimension_representation defines bottom_distance_from_datum_reference for zero, one, or more Blind_passage_template objects.

4.3.110 Blind_passage_template to Stratum_surface

Each Blind_passage_template has datum_reference defined by exactly one Stratum_surface. Each Stratum_surface defines datum_reference for zero, one, or more Blind_passage_template objects.

4.3.111 Bonded_conductive_base_blind_via to Stratum_feature_conductive_join

Each Bonded_conductive_base_blind_via has features_join defined by exactly one Stratum_feature_conductive_join. Each Stratum_feature_conductive_join defines features_join for zero or one Bonded_conductive_base_blind_via objects.

4.3.112 Bond_assembly_position to Axis_placement_3d

Each Bond_assembly_position has bond_definition_placement defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines bond_definition_placement for zero, one, or more Bond_assembly_position objects.

4.3.113 Bond_assembly_position to Default_assembly_bond_shape

Each Bond_assembly_position has bond_model defined by exactly one Default_assembly_bond_shape. Each Default_assembly_bond_shape defines bond_model for zero, one, or more Bond_assembly_position objects.

4.3.114 Bond_assembly_position to Physical_unit_3d_shape

Each Bond_assembly_position has assembly_model defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines assembly_model for zero, one, or more Bond_assembly_position objects.

4.3.115 Bond_assembly_position to Assembled_with_bonding

Each Bond_assembly_position has bond defined by exactly one Assembled_with_bonding. Each Assembled_with_bonding defines bond for zero or one Bond_assembly_position objects.

4.3.116 Boolean_csg_result to Csg_primitive

Each Boolean_csg_result has operand_2 defined by zero or one Csg_primitive objects. Each Csg_primitive defines operand_2 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type. Each Boolean_csg_result has operand_1 defined by zero or one Csg_primitive objects. Each Csg_primitive defines operand_1 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type.

4.3.117 Boolean_csg_result to Bound_volume_shape

Each Boolean_csg_result has operand_2 defined by zero or one Bound_volume_shape objects. Each Bound_volume_shape defines operand_2 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type. Each Boolean_csg_result has operand_1 defined by zero or one Bound_volume_shape objects. Each Bound_volume_shape defines operand_1 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type.

4.3.118 Boolean_csg_result to Boolean_csg_result

Each Boolean_csg_result has operand_2 defined by zero or one Boolean_csg_result objects. Each Boolean_csg_result defines operand_2 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type. Each Boolean_csg_result has operand_1 defined by zero or one Boolean_csg_result objects. Each Boolean_csg_result defines operand_1 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type.

4.3.119 Boolean_csg_result to Primitive_2d

Each Boolean_csg_result has operand_2 defined by zero or one Primitive_2d objects. Each Primitive_2d defines operand_2 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type. Each Boolean_csg_result has operand_1 defined by zero or one Primitive_2d objects. Each Primitive_2d defines operand_1 for zero, one, or more Boolean_csg_result objects. This assertion is established through a SELECT type.

4.3.120 Boundary_based_position_tolerance to Profile_of_any_surface_tolerance_with_referenced_datum_system

Each Boundary_based_position_tolerance has profile_control defined by exactly one Profile_of_any_surface_tolerance_with_referenced_datum_system. Each Profile_of_any_surface_tolerance_with_referenced_datum_system defines profile_control for zero, one, or more Boundary_based_position_tolerance objects.

4.3.121 Boundary_size_characteristic to Length_data_element

Each Boundary_size_characteristic has maximum_tolerance_value defined by exactly one Length_data_element. Each Length_data_element defines maximum_tolerance_value for zero, one, or more Boundary_size_characteristic objects. Each Boundary_size_characteristic has tolerance_value defined by exactly one Length_data_element. Each Length_data_element defines tolerance_value for zero, one, or more Boundary_size_characteristic objects.

4.3.122 Boundary_zone_definition_with_specified_size to Boundary_size_characteristic

Each Boundary_zone_definition_with_specified_size has specified_boundary_size defined by exactly one Boundary_size_characteristic. Each Boundary_size_characteristic defines specified_boundary_size for zero, one, or more Boundary_zone_definition_with_specified_size objects.

4.3.123 Bounding_curve to Trimmed_line

Each Bounding_curve has basis_curve defined by exactly one Trimmed_line. Each Trimmed_line defines basis_curve for zero, one, or more Bounding_curve objects.

4.3.124 Bounding_curve to External_definition

Each Bounding_curve has mapping_function defined by exactly one External_definition. Each External_definition defines mapping_function for zero, one, or more Bounding_curve objects.

4.3.125 **Bus_element_link to Functional_unit_network_node_definition**

Each Bus_element_link has subsequent_element defined by zero or one Functional_unit_network_node_definition objects. Each Functional_unit_network_node_definition defines subsequent_element for zero, one, or more Bus_element_link objects. This assertion is established through a SELECT type. Each Bus_element_link has precedent_element defined by zero or one Functional_unit_network_node_definition objects. Each Functional_unit_network_node_definition defines precedent_element for zero, one, or more Bus_element_link objects. This assertion is established through a SELECT type.

4.3.126 **Bus_element_link to Bus_structural_definition**

Each Bus_element_link has subsequent_element defined by zero or one Bus_structural_definition objects. Each Bus_structural_definition defines subsequent_element for zero, one, or more Bus_element_link objects. This assertion is established through a SELECT type. Each Bus_element_link has precedent_element defined by zero or one Bus_structural_definition objects. Each Bus_structural_definition defines precedent_element for zero, one, or more Bus_element_link objects. This assertion is established through a SELECT type.

4.3.127 **Bus_structural_definition to Bus_element_link**

Each Bus_structural_definition has composition defined by one or more Bus_element_link objects. Each Bus_element_link defines composition for zero, one, or more Bus_structural_definition objects.

4.3.128 **Cartesian_coordinate_system to Angle_data_element**

Each Cartesian_coordinate_system has angular_uncertainty defined by exactly one Angle_data_element. Each Angle_data_element defines angular_uncertainty for zero, one, or more Cartesian_coordinate_system objects.

4.3.129 **Cartesian_coordinate_system to Length_data_element**

Each Cartesian_coordinate_system has length_uncertainty defined by exactly one Length_data_element. Each Length_data_element defines length_uncertainty for zero, one, or more Cartesian_coordinate_system objects.

4.3.130 **Cartesian_transformation_operator_2d to Cartesian_point**

Each Cartesian_transformation_operator_2d has translation defined by exactly one Cartesian_point. Each Cartesian_point defines translation for zero, one, or more Cartesian_transformation_operator_2d objects.

4.3.131 **Cartesian_transformation_operator_2d to Orientation**

Each Cartesian_transformation_operator_2d has device_orientation defined by exactly one Orientation. Each Orientation defines device_orientation for zero, one, or more Cartesian_transformation_operator_2d objects.

4.3.132 Certification to Date_and_time

Each Certification has certification_date defined by zero or one Date_and_time objects. Each Date_and_time defines certification_date for zero, one, or more Certification objects. This assertion is established through a SELECT type.

4.3.133 Certification to Date

Each Certification has certification_date defined by zero or one Date objects. Each Date defines certification_date for zero, one, or more Certification objects. This assertion is established through a SELECT type.

4.3.134 Certification to Supplied_product_version

Each Certification has certified_product_version defined by exactly one Supplied_product_version. Each Supplied_product_version defines certified_product_version for zero or one Certification objects.

4.3.135 Certification to Ee_text

Each Certification has certification_number defined by exactly one Ee_text. Each Ee_text defines certification_number for zero, one, or more Certification objects.

4.3.136 Certification to Ee_approval

Each Certification has certification_approval defined by exactly one Ee_approval. Each Ee_approval defines certification_approval for zero, one, or more Certification objects.

4.3.137 Change_design_object_management_relationship to Managed_design_object

Each Change_design_object_management_relationship has previous_design_object defined by exactly one Managed_design_object. Each Managed_design_object defines previous_design_object for zero, one, or more Change_design_object_management_relationship objects. Each Change_design_object_management_relationship has current_design_object defined by exactly one Managed_design_object. Each Managed_design_object defines current_design_object for zero, one, or more Change_design_object_management_relationship objects.

4.3.138 Change_order to Ee_text

Each Change_order has adopted_solution defined by zero or one Ee_text objects. Each Ee_text defines adopted_solution for zero, one, or more Change_order objects. This assertion is established through a SELECT type.

4.3.139 Change_order to Design_object_management_relationship

Each Change_order has adopted_solution defined by zero or one Design_object_management_relationship objects. Each Design_object_management_relationship defines adopted_solution for zero, one, or more Change_order objects. This assertion is established through a SELECT type.

4.3.140 Change_request to Ee_text

Each Change_request has consequence defined by one or more Ee_text objects. Each Ee_text defines consequence for zero, one, or more Change_request objects. Each Change_request has recommended_solution defined by zero or one Ee_text objects. Each Ee_text defines recommended_solution for zero, one, or more Change_request objects. This assertion is established through a SELECT type.

4.3.141 Change_request to Design_object_management_relationship

Each Change_request has recommended_solution defined by zero or one Design_object_management_relationship objects. Each Design_object_management_relationship defines recommended_solution for zero, one, or more Change_request objects. This assertion is established through a SELECT type.

4.3.142 Characteristic to External_definition

Each Characteristic has property_value defined by zero or one External_definition objects. Each External_definition defines property_value for zero, one, or more Characteristic objects. This assertion is established through a SELECT type.

4.3.143 Characteristic to Compound_characteristic_definition

Each Characteristic has property_value defined by zero or one Compound_characteristic_definition objects. Each Compound_characteristic_definition defines property_value for zero, one, or more Characteristic objects. This assertion is established through a SELECT type.

4.3.144 Characteristic to Data_element

Each Characteristic has property_value defined by zero or one Data_element objects. Each Data_element defines property_value for zero, one, or more Characteristic objects. This assertion is established through a SELECT type.

4.3.145 Characteristic to Ee_document

Each Characteristic has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Characteristic objects.

4.3.146 Characteristic_data_set to Set_characteristic

Each Characteristic_data_set has property_value defined by exactly one Set_characteristic. Each Set_characteristic defines property_value for zero, one, or more Characteristic_data_set objects.

4.3.147 Characteristic_data_table to List_characteristic

Each Characteristic_data_table has property_value defined by exactly one List_characteristic. Each List_characteristic defines property_value for zero, one, or more Characteristic_data_table objects.

4.3.148 Circular_area to Length_data_element

Each Circular_area has radius defined by exactly one Length_data_element. Each Length_data_element defines radius for zero, one, or more Circular_area objects.

4.3.149 Circular_area to Cartesian_point

Each Circular_area has centre defined by exactly one Cartesian_point. Each Cartesian_point defines centre for zero, one, or more Circular_area objects.

4.3.150 Component_2d_edge_location to Assembly_joint

Each Component_2d_edge_location has mounting_surface_assembly_joint defined by exactly one Assembly_joint. Each Assembly_joint defines mounting_surface_assembly_joint for zero, one, or more Component_2d_edge_location objects. Each Component_2d_edge_location has reference_terminal_assembly_joint defined by exactly one Assembly_joint. Each Assembly_joint defines reference_terminal_assembly_joint for zero, one, or more Component_2d_edge_location objects.

4.3.151 Component_2d_embedded_location to Fabrication_joint

Each Component_2d_embedded_location has reference_terminal defined by exactly one Fabrication_joint. Each Fabrication_joint defines reference_terminal for zero, one, or more Component_2d_embedded_location objects.

4.3.152 Component_2d_embedded_location to Adjacent_stratum_surface_embedded_component_surface_definition

Each Component_2d_embedded_location has subsequent_surface defined by exactly one Adjacent_stratum_surface_embedded_component_surface_definition. Each Adjacent_stratum_surface_embedded_component_surface_definition defines subsequent_surface for zero, one, or more Component_2d_embedded_location objects. Each Component_2d_embedded_location has precedent_surface defined by exactly one Adjacent_stratum_surface_embedded_component_surface_definition. Each Adjacent_stratum_surface_embedded_component_surface_definition defines precedent_surface for zero, one, or more Component_2d_embedded_location objects.

4.3.153 Component_2d_location to Cartesian_transformation_operator_2d

Each Component_2d_location has transformation defined by exactly one Cartesian_transformation_operator_2d. Each Cartesian_transformation_operator_2d defines transformation for zero, one, or more Component_2d_location objects.

4.3.154 Component_2d_stacked_location to Assembly_joint

Each Component_2d_stacked_location has mounting_joint defined by exactly one Assembly_joint. Each Assembly_joint defines mounting_joint for zero, one, or more Component_2d_stacked_location objects.

4.3.155 Component_2d_surface_location to Assembly_joint

Each Component_2d_surface_location has mounting_surface_assembly_joint defined by exactly one Assembly_joint. Each Assembly_joint defines mounting_surface_assembly_joint for zero, one, or more Component_2d_surface_location objects.

4.3.156 Component_3d_location to Axis_placement_3d

Each Component_3d_location has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Component_3d_location objects.

4.3.157 Component_assembly_2d_position to Next_higher_assembly_relationship

Each Component_assembly_2d_position has component defined by exactly one Next_higher_assembly_relationship. Each Next_higher_assembly_relationship defines component for zero or one Component_assembly_2d_position objects.

4.3.158 Component_assembly_2d_position to Physical_unit_planar_shape

Each Component_assembly_2d_position has assembly_model defined by exactly one Physical_unit_planar_shape. Each Physical_unit_planar_shape defines assembly_model for zero, one, or more Component_assembly_2d_position objects.

4.3.159 Component_assembly_2d_position to Component_2d_location

Each Component_assembly_2d_position has transformation defined by exactly one Component_2d_location. Each Component_2d_location defines transformation for zero, one, or more Component_assembly_2d_position objects.

4.3.160 Component_assembly_2d_position to Assembly_component_2d_shape

Each Component_assembly_2d_position has component_model defined by exactly one Assembly_component_2d_shape. Each Assembly_component_2d_shape defines component_model for zero, one, or more Component_assembly_2d_position objects.

4.3.161 Component_assembly_3d_position to Next_higher_assembly_relationship

Each Component_assembly_3d_position has component defined by exactly one Next_higher_assembly_relationship. Each Next_higher_assembly_relationship defines component for zero or one Component_assembly_3d_position objects.

4.3.162 Component_assembly_3d_position to Physical_unit_3d_shape

Each Component_assembly_3d_position has assembly_model defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines assembly_model for zero, one, or more Component_assembly_3d_position objects.

4.3.163 Component_assembly_3d_position to Component_3d_location

Each Component_assembly_3d_position has transformation defined by exactly one Component_3d_location. Each Component_3d_location defines transformation for zero, one, or more Component_assembly_3d_position objects.

4.3.164 Component_assembly_3d_position to Assembly_component_3d_shape

Each Component_assembly_3d_position has component_model defined by exactly one Assembly_component_3d_shape. Each Assembly_component_3d_shape defines component_model for zero, one, or more Component_assembly_3d_position objects.

4.3.165 Component_assembly_non_planar_position to Next_higher_assembly_relationship

Each Component_assembly_non_planar_position has component defined by exactly one Next_higher_assembly_relationship. Each Next_higher_assembly_relationship defines component for zero, one, or more Component_assembly_non_planar_position objects.

4.3.166 Component_assembly_non_planar_position to Physical_unit_3d_shape

Each Component_assembly_non_planar_position has assembly_model defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines assembly_model for zero, one, or more Component_assembly_non_planar_position objects.

4.3.167 Component_assembly_non_planar_position to Component_non_planar_location

Each Component_assembly_non_planar_position has transformation defined by exactly one Component_non_planar_location. Each Component_non_planar_location defines transformation for zero, one, or more Component_assembly_non_planar_position objects.

4.3.168 Component_assembly_non_planar_position to Assembly_component_non_planar_shape

Each Component_assembly_non_planar_position has component_model defined by exactly one Assembly_component_non_planar_shape. Each Assembly_component_non_planar_shape defines component_model for zero, one, or more Component_assembly_non_planar_position objects.

4.3.169 Component_based_annotation_symbol_placement_relationship to Special_symbol_laminate_component

Each Component_based_annotation_symbol_placement_relationship has related_stratum_concept defined by exactly one Special_symbol_laminate_component. Each Special_symbol_laminate_component defines related_stratum_concept for zero, one, or more Component_based_annotation_symbol_placement_relationship objects.

4.3.170 Component_based_annotation_text_relationship to Laminate_text_string_component

Each Component_based_annotation_text_relationship has related_stratum_concept defined by exactly one Laminate_text_string_component. Each Laminate_text_string_component defines related_stratum_concept for zero, one, or more Component_based_annotation_text_relationship objects.

4.3.171 Component_feature to Assembly_component

Each Component_feature has associated_component defined by exactly one Assembly_component. Each Assembly_component defines associated_component for zero, one, or more Component_feature objects.

4.3.172 Component_feature to Land_template_terminal

Each Component_feature has definition defined by zero or one Land_template_terminal objects. Each Land_template_terminal defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.173 Component_feature to Component_termination_passage_template_terminal

Each Component_feature has definition defined by zero or one Component_termination_passage_template_terminal objects. Each Component_termination_passage_template_terminal defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.174 Component_feature to Part_feature

Each Component_feature has definition defined by zero or one Part_feature objects. Each Part_feature defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.175 Component_feature to Printed_part_template_terminal

Each Component_feature has definition defined by zero or one Printed_part_template_terminal objects. Each Printed_part_template_terminal defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.176 Component_feature to Part_terminal_external_reference

Each Component_feature has definition defined by zero or one Part_terminal_external_reference objects. Each Part_terminal_external_reference defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.177 Component_feature to Via_template_terminal

Each Component_feature has definition defined by zero or one Via_template_terminal objects. Each Via_template_terminal defines definition for zero, one, or more Component_feature objects. This assertion is established through a SELECT type.

4.3.178 Component_feature_external_reference to Component_external_reference

Each Component_feature_external_reference has associated_component defined by exactly one Component_external_reference. Each Component_external_reference defines associated_component for zero, one, or more Component_feature_external_reference objects.

4.3.179 Component_feature_make_from_relationship to Component_feature

Each Component_feature_make_from_relationship has resultant_feature defined by exactly one Component_feature. Each Component_feature defines resultant_feature for zero, one, or more Component_feature_make_from_relationship objects. Each Component_feature_make_from_relationship has reusable_feature defined by exactly one Component_feature. Each Component_feature defines reusable_feature for zero, one, or more Component_feature_make_from_relationship objects.

4.3.180 Component_feature_relationship to Component_feature

Each Component_feature_relationship has relating_component_feature defined by exactly one Component_feature. Each Component_feature defines relating_component_feature for zero, one, or more Component_feature_relationship objects. Each Component_feature_relationship has related_component_feature defined by exactly one Component_feature. Each Component_feature defines related_component_feature for zero, one, or more Component_feature_relationship objects.

4.3.181 Component_feature_to_layout_feature_relationship to Ee_requirement_occurrence

Each Component_feature_to_layout_feature_relationship has requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines requirement for zero, one, or more Component_feature_to_layout_feature_relationship objects.

4.3.182 Component_feature_to_layout_feature_relationship to Assembly_component

Each Component_feature_to_layout_feature_relationship has relating_feature defined by zero or one Assembly_component objects. Each Assembly_component defines relating_feature for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.183 Component_feature_to_layout_feature_relationship to Component_external_reference

Each Component_feature_to_layout_feature_relationship has relating_feature defined by zero or one Component_external_reference objects. Each Component_external_reference defines relating_feature for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.184 Component_feature_to_layout_feature_relationship to Component_feature

Each Component_feature_to_layout_feature_relationship has relating_feature defined by zero or one Component_feature objects. Each Component_feature defines relating_feature for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.185 Component_feature_to_layout_feature_relationship to Component_feature_external_reference

Each Component_feature_to_layout_feature_relationship has relating_feature defined by zero or one Component_feature_external_reference objects. Each Component_feature_external_reference defines relating_feature for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.186 Component_feature_to_layout_feature_relationship to Laminate_component

Each Component_feature_to_layout_feature_relationship has related_stratum_concept defined by zero or one Laminate_component objects. Each Laminate_component defines related_stratum_concept for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.187 Component_feature_to_layout_feature_relationship to Inter_stratum_feature

Each Component_feature_to_layout_feature_relationship has related_stratum_concept defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines related_stratum_concept for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.188 Component_feature_to_layout_feature_relationship to Stratum_feature

Each Component_feature_to_layout_feature_relationship has related_stratum_concept defined by zero or one Stratum_feature objects. Each Stratum_feature defines related_stratum_concept for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.189 Component_feature_to_layout_feature_relationship to Stratum

Each Component_feature_to_layout_feature_relationship has related_stratum_concept defined by zero or one Stratum objects. Each Stratum defines related_stratum_concept for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.190 Component_feature_to_layout_feature_relationship to Stratum_surface

Each Component_feature_to_layout_feature_relationship has related_stratum_concept defined by zero or one Stratum_surface objects. Each Stratum_surface defines related_stratum_concept for zero, one, or more Component_feature_to_layout_feature_relationship objects. This assertion is established through a SELECT type.

4.3.191 Component_feature_to_physical_usage_view_assignment to Component_feature

Each Component_feature_to_physical_usage_view_assignment has assigned_design_object defined by exactly one Component_feature. Each Component_feature defines assigned_design_object for zero or one Component_feature_to_physical_usage_view_assignment objects.

4.3.192 Component_feature_to_physical_usage_view_assignment to Part_feature

Each Component_feature_to_physical_usage_view_assignment has defined_part_feature defined by exactly one Part_feature. Each Part_feature defines defined_part_feature for zero or one Component_feature_to_physical_usage_view_assignment objects.

4.3.193 Component_group_assignment to Assembly_component

Each Component_group_assignment has assigned_component defined by exactly one Assembly_component. Each Assembly_component defines assigned_component for zero, one, or more Component_group_assignment objects.

4.3.194 Component_group_assignment to Placement_group

Each Component_group_assignment has associated_group defined by exactly one Placement_group. Each Placement_group defines associated_group for zero, one, or more Component_group_assignment objects.

4.3.195 Component_interface_terminal to Interface_component

Each Component_interface_terminal has associated_component defined by exactly one Interface_component. Each Interface_component defines associated_component for one or more Component_interface_terminal objects.

4.3.196 Component_make_from_relationship to Assembly_component

Each Component_make_from_relationship has resultant_component defined by exactly one Assembly_component. Each Assembly_component defines resultant_component for zero, one, or more Component_make_from_relationship objects. Each Component_make_from_relationship has reusable_component defined by exactly one Assembly_component. Each Assembly_component defines reusable_component for zero, one, or more Component_make_from_relationship objects.

4.3.197 Component_mounting_clearance_relationship to Length_data_element

Each Component_mounting_clearance_relationship has maximum_mounting_clearance defined by exactly one Length_data_element. Each Length_data_element defines maximum_mounting_clearance for zero, one, or more Component_mounting_clearance_relationship objects. Each Component_mounting_clearance_relationship has minimum_mounting_clearance defined by exactly one Length_data_element. Each Length_data_element defines minimum_mounting_clearance for zero, one, or more Component_mounting_clearance_relationship objects.

4.3.198 Component_mounting_clearance_relationship to Component_feature_relationship

Each Component_mounting_clearance_relationship has feature_relationship defined by exactly one Component_feature_relationship. Each Component_feature_relationship defines feature_relationship for zero, one, or more Component_mounting_clearance_relationship objects.

4.3.199 Component_mounting_feature to Part_mounting_feature

Each Component_mounting_feature has definition defined by exactly one Part_mounting_feature. Each Part_mounting_feature defines definition for zero, one, or more Component_mounting_feature objects.

4.3.200 Component_non_planar_location to Axis_placement_3d

Each Component_non_planar_location has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Component_non_planar_location objects.

4.3.201 Component_overlap_relationship to Length_data_element

Each Component_overlap_relationship has maximum_added_clearance defined by exactly one Length_data_element. Each Length_data_element defines maximum_added_clearance for zero, one, or more Component_overlap_relationship objects. Each Component_overlap_relationship has minimum_added_clearance defined by exactly one Length_data_element. Each Length_data_element defines minimum_added_clearance for zero, one, or more Component_overlap_relationship objects.

4.3.202 Component_overlap_relationship to Next_higher_assembly_relationship

Each Component_overlap_relationship has current_component defined by exactly one Next_higher_assembly_relationship. Each Next_higher_assembly_relationship defines current_component for zero, one, or more Component_overlap_relationship objects. Each Component_overlap_relationship has previously_placed_component defined by exactly one Next_higher_assembly_relationship. Each Next_higher_assembly_relationship defines previously_placed_component for zero, one, or more Component_overlap_relationship objects.

4.3.203 Component_placement_restriction_assignment to Datum_based_length_measure

Each Component_placement_restriction_assignment has minimum_mounting_clearance defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines minimum_mounting_clearance for zero, one, or more Component_placement_restriction_assignment objects. Each Component_placement_restriction_assignment has maximum_mounting_clearance defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines maximum_mounting_clearance for zero, one, or more Component_placement_restriction_assignment objects. Each Component_placement_restriction_assignment has maximum_positive_component_height defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines maximum_positive_component_height for zero, one, or more Component_placement_restriction_assignment objects. Each Component_placement_restriction_assignment has maximum_negative_component_height defined by exactly one Datum_based_length_measure. Each Datum_based_length_measure defines maximum_negative_component_height for zero, one, or more Component_placement_restriction_assignment objects.

4.3.204 Component_placement_restriction_assignment to Item_restricted_requirement_occurrence

Each Component_placement_restriction_assignment has requirement defined by exactly one Item_restricted_requirement_occurrence. Each Item_restricted_requirement_occurrence defines requirement for zero, one, or more Component_placement_restriction_assignment objects.

4.3.205 Component_placement_restriction_assignment to Mounting_restriction_volume

Each Component_placement_restriction_assignment has volume defined by exactly one Mounting_restriction_volume. Each Mounting_restriction_volume defines volume for zero, one, or more Component_placement_restriction_assignment objects.

4.3.206 Component_placement_restriction_assignment to Mounting_restriction_area

Each Component_placement_restriction_assignment has area defined by exactly one Mounting_restriction_area. Each Mounting_restriction_area defines area for zero, one, or more Component_placement_restriction_assignment objects.

4.3.207 Component_swap_specification to Design_functional_unit_allocation_to_assembly_component

Each Component_swap_specification has previous_design_object defined by exactly one Design_functional_unit_allocation_to_assembly_component. Each Design_functional_unit_allocation_to_assembly_component defines previous_design_object for zero, one, or more Component_swap_specification objects. Each Component_swap_specification has current_design_object defined by exactly one Design_functional_unit_allocation_to_assembly_component. Each Design_functional_unit_allocation_to_assembly_component defines current_design_object for zero, one, or more Component_swap_specification objects.

4.3.208 Component_termination_passage to Component_termination_passage_template

Each Component_termination_passage has usage_definition defined by exactly one Component_termination_passage_template. Each Component_termination_passage_template defines usage_definition for zero, one, or more Component_termination_passage objects.

4.3.209 Component_termination_passage_and_contact_size_dependent_land to Component_termination_passage

Each Component_termination_passage_and_contact_size_dependent_land has reference_passage defined by exactly one Component_termination_passage. Each Component_termination_passage defines reference_passage for zero, one, or more Component_termination_passage_and_contact_size_dependent_land objects.

4.3.210 Component_termination_passage_and_contact_size_dependent_land to Default_attachment_size_and_component_termination_passage_based_land_physical_template

Each Component_termination_passage_and_contact_size_dependent_land has usage_definition defined by exactly one Default_attachment_size_and_component_termination_passage_based_land_physical_template. Each Default_attachment_size_and_component_termination_passage_based_land_physical_template defines usage_definition for zero, one, or more Component_termination_passage_and_contact_size_dependent_land objects.

4.3.211**Component_termination_passage_and_contact_size_dependent_non_functional_land to Component_termination_passage**

Each Component_termination_passage_and_contact_size_dependent_non_functional_land has reference_passage defined by exactly one Component_termination_passage. Each Component_termination_passage defines reference_passage for zero, one, or more Component_termination_passage_and_contact_size_dependent_non_functional_land objects.

4.3.212**Component_termination_passage_and_contact_size_dependent_non_functional_land to Default_attachment_size_and_component_termination_passage_based_land_physical_template**

Each Component_termination_passage_and_contact_size_dependent_non_functional_land has usage_definition defined by exactly one Default_attachment_size_and_component_termination_passage_based_land_physical_template. Each Default_attachment_size_and_component_termination_passage_based_land_physical_template defines usage_definition for zero, one, or more Component_termination_passage_and_contact_size_dependent_non_functional_land objects.

4.3.213 Component_termination_passage_dependent_land to Default_component_termination_passage_based_land_physical_template

Each Component_termination_passage_dependent_land has usage_definition defined by exactly one Default_component_termination_passage_based_land_physical_template. Each Default_component_termination_passage_based_land_physical_template defines usage_definition for zero, one, or more Component_termination_passage_dependent_land objects.

4.3.214 Component_termination_passage_dependent_land to Component_termination_passage

Each Component_termination_passage_dependent_land has reference_passage defined by exactly one Component_termination_passage. Each Component_termination_passage defines reference_passage for zero, one, or more Component_termination_passage_dependent_land objects.

4.3.215 Component_termination_passage_dependent_non_functional_land to Default_component_termination_passage_based_land_physical_template

Each Component_termination_passage_dependent_non_functional_land has usage_definition defined by exactly one Default_component_termination_passage_based_land_physical_template. Each Default_component_termination_passage_based_land_physical_template defines usage_definition for zero, one, or more Component_termination_passage_dependent_non_functional_land objects.

4.3.216 Component_termination_passage_dependent_non_functional_land to Component_termination_passage

Each Component_termination_passage_dependent_non_functional_land has reference_passage defined by exactly one Component_termination_passage. Each Component_termination_passage defines reference_passage for zero, one, or more Component_termination_passage_dependent_non_functional_land objects.

4.3.217 Component_termination_passage_interface_terminal to Component_termination_passage

Each Component_termination_passage_interface_terminal has associated_component defined by exactly one Component_termination_passage. Each Component_termination_passage defines associated_component for zero, one, or more Component_termination_passage_interface_terminal objects.

4.3.218 Component_termination_passage_interface_terminal to Component_termination_passage_template_interface_terminal

Each Component_termination_passage_interface_terminal has definition defined by exactly one Component_termination_passage_template_interface_terminal. Each Component_termination_passage_template_interface_terminal defines definition for zero, one, or more Component_termination_passage_interface_terminal objects.

4.3.219 Component_termination_passage_join_terminal to Component_termination_passage

Each Component_termination_passage_join_terminal has associated_component defined by exactly one Component_termination_passage. Each Component_termination_passage defines associated_component for one or more Component_termination_passage_join_terminal objects.

4.3.220 Component_termination_passage_join_terminal to Component_termination_passage_template_join_terminal

Each Component_termination_passage_join_terminal has definition defined by exactly one Component_termination_passage_template_join_terminal. Each Component_termination_passage_template_join_terminal defines definition for zero, one, or more Component_termination_passage_join_terminal objects.

4.3.221 Component_termination_passage_template to Default_component_termination_passage_definition

Each Component_termination_passage_template has of_passage_technology defined by exactly one Default_component_termination_passage_definition. Each Default_component_termination_passage_definition defines of_passage_technology for zero, one, or more Component_termination_passage_template objects.

4.3.222 Component_termination_passage_template_join_terminal to Inter_stratum_extent

Each Component_termination_passage_template_join_terminal has disallowed_inter_stratum_extent defined by one or more Inter_stratum_extent objects. Each Inter_stratum_extent defines disallowed_inter_stratum_extent for zero, one, or more Component_termination_passage_template_join_terminal objects.

4.3.223 Component_termination_passage_template_terminal to Connection_zone

Each Component_termination_passage_template_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Component_termination_passage_template_terminal objects.

4.3.224 Component_termination_passage_template_terminal to Component_termination_passage_template

Each Component_termination_passage_template_terminal has associated_definition defined by exactly one Component_termination_passage_template. Each Component_termination_passage_template defines associated_definition for two or more Component_termination_passage_template_terminal objects.

4.3.225 Composite_curve to Composite_curve_segment

Each Composite_curve has curve_segment defined by one or more Composite_curve_segment objects. Each Composite_curve_segment defines curve_segment for one or more Composite_curve objects.

4.3.226 Composite_curve_segment to Curve

Each Composite_curve_segment has segment_curve defined by exactly one Curve. Each Curve defines segment_curve for zero, one, or more Composite_curve_segment objects.

4.3.227 Composite_feature_relating_position_tolerance to Composite_feature_locating_position_tolerance

Each Composite_feature_relating_position_tolerance has refined_control defined by zero or one Composite_feature_locating_position_tolerance objects. Each Composite_feature_locating_position_tolerance defines refined_control for zero, one, or more Composite_feature_relating_position_tolerance objects. This assertion is established through a SELECT type.

4.3.228 Composite_feature_relating_position_tolerance to Composite_feature_relating_position_tolerance

Each Composite_feature_relating_position_tolerance has refined_control defined by zero or one Composite_feature_relating_position_tolerance objects. Each Composite_feature_relating_position_tolerance defines refined_control for zero, one, or more Composite_feature_relating_position_tolerance objects. This assertion is established through a SELECT type.

4.3.229 Composite_linear_profile_refinement_tolerance to Composite_linear_profile_locating_tolerance

Each Composite_linear_profile_refinement_tolerance has refined_control defined by zero or one Composite_linear_profile_locating_tolerance objects. Each Composite_linear_profile_locating_tolerance defines refined_control for zero, one, or more Composite_linear_profile_refinement_tolerance objects. This assertion is established through a SELECT type.

4.3.230 Composite_linear_profile_refinement_tolerance to Composite_linear_profile_refinement_tolerance

Each Composite_linear_profile_refinement_tolerance has refined_control defined by zero or one Composite_linear_profile_refinement_tolerance objects. Each Composite_linear_profile_refinement_tolerance defines refined_control for zero, one, or more Composite_linear_profile_refinement_tolerance objects. This assertion is established through a SELECT type.

4.3.231 Composite_signal_property_relationship to External_definition

Each Composite_signal_property_relationship has aggregation_operation defined by exactly one External_definition. Each External_definition defines aggregation_operation for zero, one, or more Composite_signal_property_relationship objects.

4.3.232 Composite_signal_property_relationship to Signal_property

Each Composite_signal_property_relationship has signal_property_structural_element defined by exactly one Signal_property. Each Signal_property defines signal_property_structural_element for zero, one, or more Composite_signal_property_relationship objects.

4.3.233 Composite_signal_property_relationship to Composite_signal_property

Each Composite_signal_property_relationship has composition defined by exactly one Composite_signal_property. Each Composite_signal_property defines composition for one or more Composite_signal_property_relationship objects.

4.3.234 Composite_surface_profile_refinement_tolerance to Composite_surface_profile_locating_tolerance

Each Composite_surface_profile_refinement_tolerance has refined_control defined by zero or one Composite_surface_profile_locating_tolerance objects. Each Composite_surface_profile_locating_tolerance defines refined_control for zero, one, or more Composite_surface_profile_refinement_tolerance objects. This assertion is established through a SELECT type.

4.3.235 Composite_surface_profile_refinement_tolerance to Composite_surface_profile_refinement_tolerance

Each Composite_surface_profile_refinement_tolerance has refined_control defined by zero or one Composite_surface_profile_refinement_tolerance objects. Each Composite_surface_profile_refinement_tolerance defines refined_control for zero, one, or more Composite_surface_profile_refinement_tolerance objects. This assertion is established through a SELECT type.

4.3.236 Compound_characteristic to Compound_characteristic_definition

Each Compound_characteristic has property_value defined by exactly one Compound_characteristic_definition. Each Compound_characteristic_definition defines property_value for zero, one, or more Compound_characteristic objects.

4.3.237 Conductive_interconnect_element to Stratum_feature

Each Conductive_interconnect_element has composed_conductor defined by exactly one Stratum_feature. Each Stratum_feature defines composed_conductor for zero, one, or more Conductive_interconnect_element objects.

4.3.238 Conductive_interconnect_element_terminal to Connection_zone

Each Conductive_interconnect_element_terminal has connection defined by exactly one Connection_zone. Each Connection_zone defines connection for zero, one, or more Conductive_interconnect_element_terminal objects.

4.3.239 Conductive_interconnect_element_terminal to Conductive_interconnect_element

Each Conductive_interconnect_element_terminal has associated_component defined by exactly one Conductive_interconnect_element. Each Conductive_interconnect_element defines associated_component for exactly two Conductive_interconnect_element_terminal objects.

4.3.240 Conductive_interconnect_element_with_pre_defined_transitions to Layer_connection_point_link

Each Conductive_interconnect_element_with_pre_defined_transitions has connected_points defined by one or more Layer_connection_point_link objects. Each Layer_connection_point_link defines connected_points for zero, one, or more Conductive_interconnect_element_with_pre_defined_transitions objects.

4.3.241 Conductive_interconnect_element_with_pre_defined_transitions to Curve

Each Conductive_interconnect_element_with_pre_defined_transitions has centreline_shape defined by exactly one Curve. Each Curve defines centreline_shape for zero, one, or more Conductive_interconnect_element_with_pre_defined_transitions objects.

4.3.242 Conductive_interconnect_element_with_pre_defined_transitions to Trace_template

Each `Conductive_interconnect_element_with_pre_defined_transitions` has `usage_definition` defined by exactly one `Trace_template`. Each `Trace_template` defines `usage_definition` for zero, one, or more `Conductive_interconnect_element_with_pre_defined_transitions` objects.

4.3.243 Conductive_interconnect_element_with_user_defined_single_transition to Layer_connection_point

Each `Conductive_interconnect_element_with_user_defined_single_transition` has `end_terminus` defined by exactly one `Layer_connection_point`. Each `Layer_connection_point` defines `end_terminus` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects. Each `Conductive_interconnect_element_with_user_defined_single_transition` has `start_terminus` defined by exactly one `Layer_connection_point`. Each `Layer_connection_point` defines `start_terminus` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects.

4.3.244 Conductive_interconnect_element_with_user_defined_single_transition to Bounding_curve

Each `Conductive_interconnect_element_with_user_defined_single_transition` has `right_curve_shape_representation` defined by zero or one `Bounding_curve` objects. Each `Bounding_curve` defines `right_curve_shape_representation` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects. This assertion is established through a `SELECT` type. Each `Conductive_interconnect_element_with_user_defined_single_transition` has `left_curve_shape_representation` defined by zero or one `Bounding_curve` objects. Each `Bounding_curve` defines `left_curve_shape_representation` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects. This assertion is established through a `SELECT` type.

4.3.245 Conductive_interconnect_element_with_user_defined_single_transition to Trimmed_line

Each `Conductive_interconnect_element_with_user_defined_single_transition` has `right_curve_shape_representation` defined by zero or one `Trimmed_line` objects. Each `Trimmed_line` defines `right_curve_shape_representation` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects. This assertion is established through a `SELECT` type. Each `Conductive_interconnect_element_with_user_defined_single_transition` has `left_curve_shape_representation` defined by zero or one `Trimmed_line` objects. Each `Trimmed_line` defines `left_curve_shape_representation` for zero, one, or more `Conductive_interconnect_element_with_user_defined_single_transition` objects. This assertion is established through a `SELECT` type.

4.3.246 Conductive_interconnect_element_with_user_defined_single_transition to Trace_template

Each Conductive_interconnect_element_with_user_defined_single_transition has end_definition defined by exactly one Trace_template. Each Trace_template defines end_definition for zero, one, or more Conductive_interconnect_element_with_user_defined_single_transition objects. Each Conductive_interconnect_element_with_user_defined_single_transition has start_definition defined by exactly one Trace_template. Each Trace_template defines start_definition for zero, one, or more Conductive_interconnect_element_with_user_defined_single_transition objects.

4.3.247 Conductor to Intra_stratum_join_relationship

Each Conductor has implementation_requirement defined by exactly one Intra_stratum_join_relationship. Each Intra_stratum_join_relationship defines implementation_requirement for zero or one Conductor objects.

4.3.248 Conductor to Design_layer_stratum

Each Conductor has resident_stratum defined by exactly one Design_layer_stratum. Each Design_layer_stratum defines resident_stratum for zero, one, or more Conductor objects.

4.3.249 Configuration_controlled_message to Organization

Each Configuration_controlled_message has message_source defined by zero or one Organization objects. Each Organization defines message_source for zero, one, or more Configuration_controlled_message objects. This assertion is established through a SELECT type.

4.3.250 Configuration_controlled_message to Ee_approval

Each Configuration_controlled_message has message_source defined by zero or one Ee_approval objects. Each Ee_approval defines message_source for zero, one, or more Configuration_controlled_message objects. This assertion is established through a SELECT type.

4.3.251 Configuration_controlled_message to Ee_product_version

Each Configuration_controlled_message has message_source defined by zero or one Ee_product_version objects. Each Ee_product_version defines message_source for zero, one, or more Configuration_controlled_message objects. This assertion is established through a SELECT type.

4.3.252 Conical_tolerance_zone_boundary_and_surface_relationship to Physical_feature_or_part_template

Each Conical_tolerance_zone_boundary_and_surface_relationship has terminating_surface defined by exactly one Physical_feature_or_part_template. Each Physical_feature_or_part_template defines terminating_surface for zero, one, or more Conical_tolerance_zone_boundary_and_surface_relationship objects.

4.3.253 Conical_tolerance_zone_boundary_and_surface_relationship to Conical_tolerance_zone_boundary

Each Conical_tolerance_zone_boundary_and_surface_relationship has sized_tolerance_zone_boundary defined by exactly one Conical_tolerance_zone_boundary. Each Conical_tolerance_zone_boundary defines sized_tolerance_zone_boundary for exactly two Conical_tolerance_zone_boundary_and_surface_relationship objects.

4.3.254 Conical_tolerance_zone_boundary_definition to Conical_tolerance_zone_boundary

Each Conical_tolerance_zone_boundary_definition has defined_zone_boundary defined by exactly one Conical_tolerance_zone_boundary. Each Conical_tolerance_zone_boundary defines defined_zone_boundary for zero, one, or more Conical_tolerance_zone_boundary_definition objects.

4.3.255 Conical_tolerance_zone_boundary_definition to Conical_tolerance_zone_boundary_diametrical_size_characteristic

Each Conical_tolerance_zone_boundary_definition has boundary_sizes defined by exactly two Conical_tolerance_zone_boundary_diametrical_size_characteristic objects. Each Conical_tolerance_zone_boundary_diametrical_size_characteristic defines boundary_sizes for zero, one, or more Conical_tolerance_zone_boundary_definition objects.

4.3.256 Conical_tolerance_zone_boundary_diametrical_size_characteristic to Length_data_element

Each Conical_tolerance_zone_boundary_diametrical_size_characteristic has tolerance_value defined by exactly one Length_data_element. Each Length_data_element defines tolerance_value for zero, one, or more Conical_tolerance_zone_boundary_diametrical_size_characteristic objects.

4.3.257 Conical_tolerance_zone_boundary_diametrical_size_characteristic to Conical_tolerance_zone_boundary_and_surface_relationship

Each Conical_tolerance_zone_boundary_diametrical_size_characteristic has sized_boundary_at_surface defined by exactly one Conical_tolerance_zone_boundary_and_surface_relationship. Each Conical_tolerance_zone_boundary_and_surface_relationship defines sized_boundary_at_surface for zero, one, or more Conical_tolerance_zone_boundary_diametrical_size_characteristic objects.

4.3.258 Conical_tolerance_zone_definition to Conical_tolerance_zone_boundary_definition

Each Conical_tolerance_zone_definition has boundary_definition defined by exactly one Conical_tolerance_zone_boundary_definition. Each Conical_tolerance_zone_boundary_definition defines boundary_definition for zero, one, or more Conical_tolerance_zone_definition objects.

4.3.259 Connected_area_component to Layer_connection_point

Each Connected_area_component has associated_layer_connection_point defined by one or more Layer_connection_point objects. Each Layer_connection_point defines associated_layer_connection_point for zero, one, or more Connected_area_component objects.

4.3.260 Connected_filled_area to Intra_stratum_join_relationship

Each Connected_filled_area has implementation_requirement defined by exactly one Intra_stratum_join_relationship. Each Intra_stratum_join_relationship defines implementation_requirement for zero or one Connected_filled_area objects.

4.3.261 Connected_filled_area to Design_layer_stratum

Each Connected_filled_area has resident_stratum defined by exactly one Design_layer_stratum. Each Design_layer_stratum defines resident_stratum for zero, one, or more Connected_filled_area objects.

4.3.262 Connection_zone to Ee_colour

Each Connection_zone has representation_colour defined by exactly one Ee_colour. Each Ee_colour defines representation_colour for zero, one, or more Connection_zone objects.

4.3.263 Connection_zone_based_assembly_joint to Connection_zone

Each Connection_zone_based_assembly_joint has zone_2 defined by exactly one Connection_zone. Each Connection_zone defines zone_2 for zero, one, or more Connection_zone_based_assembly_joint objects. Each Connection_zone_based_assembly_joint has zone_1 defined by exactly one Connection_zone. Each Connection_zone defines zone_1 for zero, one, or more Connection_zone_based_assembly_joint objects.

4.3.264 Connection_zone_interface_plane_relationship to Connection_zone

Each Connection_zone_interface_plane_relationship has derived_zone defined by exactly one Connection_zone. Each Connection_zone defines derived_zone for zero, one, or more Connection_zone_interface_plane_relationship objects. Each Connection_zone_interface_plane_relationship has mating_zone defined by exactly one Connection_zone. Each Connection_zone defines mating_zone for zero, one, or more Connection_zone_interface_plane_relationship objects.

4.3.265 Connection_zone_interface_plane_relationship to Seating_plane

Each Connection_zone_interface_plane_relationship has interface_plane defined by exactly one Seating_plane. Each Seating_plane defines interface_plane for zero, one, or more Connection_zone_interface_plane_relationship objects.

4.3.266 Connection_zone_map_identification to Usage_view_connection_zone_terminal_shape_relationship

Each Connection_zone_map_identification has mapped_connection_zone_shape defined by exactly one Usage_view_connection_zone_terminal_shape_relationship. Each Usage_view_connection_zone_terminal_shape_relationship defines mapped_connection_zone_shape for zero, one, or more Connection_zone_map_identification objects. Each Connection_zone_map_identification has reference_connection_zone_shape defined by exactly one Usage_view_connection_zone_terminal_shape_relationship. Each Usage_view_connection_zone_terminal_shape_relationship defines reference_connection_zone_shape for zero, one, or more Connection_zone_map_identification objects.

4.3.267 Connection_zone_model_port_assignment to Analytical_model_port

Each Connection_zone_model_port_assignment has target defined by exactly one Analytical_model_port. Each Analytical_model_port defines target for zero, one, or more Connection_zone_model_port_assignment objects.

4.3.268 Connection_zone_model_port_assignment to Connection_zone

Each Connection_zone_model_port_assignment has source defined by exactly one Connection_zone. Each Connection_zone defines source for zero, one, or more Connection_zone_model_port_assignment objects.

4.3.269 Connectivity_allocation to Aggregate_connectivity_requirement

Each Connectivity_allocation has connectivity_requirement defined by exactly one Aggregate_connectivity_requirement. Each Aggregate_connectivity_requirement defines connectivity_requirement for zero, one, or more Connectivity_allocation objects.

4.3.270 Connectivity_allocation to Physical_connectivity_definition

Each Connectivity_allocation has implementation defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines implementation for zero, one, or more Connectivity_allocation objects.

4.3.271 Connectivity_allocation_to_physical_network to Aggregate_connectivity_requirement

Each Connectivity_allocation_to_physical_network has connectivity_requirement defined by exactly one Aggregate_connectivity_requirement. Each Aggregate_connectivity_requirement defines connectivity_requirement for zero, one, or more Connectivity_allocation_to_physical_network objects.

4.3.272 Connectivity_allocation_to_physical_network to Physical_network

Each Connectivity_allocation_to_physical_network has implementation defined by exactly one Physical_network. Each Physical_network defines implementation for zero, one, or more Connectivity_allocation_to_physical_network objects.

4.3.273 Contact_size_dependent_land to Default_attachment_size_based_land_physical_template

Each Contact_size_dependent_land has usage_definition defined by exactly one Default_attachment_size_based_land_physical_template. Each Default_attachment_size_based_land_physical_template defines usage_definition for zero, one, or more Contact_size_dependent_land objects.

4.3.274 Contact_size_dependent_non_functional_land to Default_attachment_size_based_land_physical_template

Each Contact_size_dependent_non_functional_land has usage_definition defined by exactly one Default_attachment_size_based_land_physical_template. Each Default_attachment_size_based_land_physical_template defines usage_definition for zero, one, or more Contact_size_dependent_non_functional_land objects.

4.3.275 Contract to Date_and_time

Each Contract has contract_date defined by zero or one Date_and_time objects. Each Date_and_time defines contract_date for zero, one, or more Contract objects. This assertion is established through a SELECT type.

4.3.276 Contract to Date

Each Contract has contract_date defined by zero or one Date objects. Each Date defines contract_date for zero, one, or more Contract objects. This assertion is established through a SELECT type.

4.3.277 Contract to Ee_approval

Each Contract has contract_approval defined by exactly one Ee_approval. Each Ee_approval defines contract_approval for zero, one, or more Contract objects.

4.3.278 Contract to Ee_text

Each Contract has contract_number defined by exactly one Ee_text. Each Ee_text defines contract_number for zero, one, or more Contract objects.

4.3.279 Contract to Person_and_organization

Each Contract has contractor defined by zero or one Person_and_organization objects. Each Person_and_organization defines contractor for zero, one, or more Contract objects. This assertion is established through a SELECT type.

4.3.280 Contract to Organization

Each Contract has contractor defined by zero or one Organization objects. Each Organization defines contractor for zero, one, or more Contract objects. This assertion is established through a SELECT type.

4.3.281 Coordinated_characteristic to Ee_document

Each Coordinated_characteristic has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Coordinated_characteristic objects.

4.3.282 Coordinated_characteristic to Characteristic

Each Coordinated_characteristic has component_characteristic defined by one or more Characteristic objects. Each Characteristic defines component_characteristic for zero, one, or more Coordinated_characteristic objects.

4.3.283 Counterbore_passage_template to Blind_passage_template

Each Counterbore_passage_template has larger_passage defined by exactly one Blind_passage_template. Each Blind_passage_template defines larger_passage for zero, one, or more Counterbore_passage_template objects.

4.3.284 Counterbore_passage_template to Unsupported_passage_template

Each Counterbore_passage_template has smaller_passage defined by exactly one Unsupported_passage_template. Each Unsupported_passage_template defines smaller_passage for zero, one, or more Counterbore_passage_template objects.

4.3.285 Countersunk_passage_template to Blind_passage_template

Each Countersunk_passage_template has tapered_passage defined by exactly one Blind_passage_template. Each Blind_passage_template defines tapered_passage for zero, one, or more Countersunk_passage_template objects.

4.3.286 Countersunk_passage_template to Unsupported_passage_template

Each Countersunk_passage_template has constant_diameter_passage defined by exactly one Unsupported_passage_template. Each Unsupported_passage_template defines constant_diameter_passage for zero, one, or more Countersunk_passage_template objects.

4.3.287 Curve_style to Curve_style_font_and_scaling

Each Curve_style has curve_font defined by zero or one Curve_style_font_and_scaling objects. Each Curve_style_font_and_scaling defines curve_font for zero, one, or more Curve_style objects. This assertion is established through a SELECT type.

4.3.288 Curve_style to Solid_curve

Each Curve_style has curve_font defined by zero or one Solid_curve objects. Each Solid_curve defines curve_font for zero, one, or more Curve_style objects. This assertion is established through a SELECT type.

4.3.289 Curve_style to Externally_defined_curve_font

Each Curve_style has curve_font defined by zero or one Externally_defined_curve_font objects. Each Externally_defined_curve_font defines curve_font for zero, one, or more Curve_style objects. This assertion is established through a SELECT type.

4.3.290 Curve_style to Curve_style_font

Each Curve_style has curve_font defined by zero or one Curve_style_font objects. Each Curve_style_font defines curve_font for zero, one, or more Curve_style objects. This assertion is established through a SELECT type.

4.3.291 Curve_style to Length_data_element

Each Curve_style has curve_width defined by exactly one Length_data_element. Each Length_data_element defines curve_width for zero, one, or more Curve_style objects.

4.3.292 Curve_style_font_and_scaling to Solid_curve

Each Curve_style_font_and_scaling has curve_font defined by zero or one Solid_curve objects. Each Solid_curve defines curve_font for zero, one, or more Curve_style_font_and_scaling objects. This assertion is established through a SELECT type.

4.3.293 Curve_style_font_and_scaling to Externally_defined_curve_font

Each Curve_style_font_and_scaling has curve_font defined by zero or one Externally_defined_curve_font objects. Each Externally_defined_curve_font defines curve_font for zero, one, or more Curve_style_font_and_scaling objects. This assertion is established through a SELECT type.

4.3.294 Curve_style_font_and_scaling to Curve_style_font

Each Curve_style_font_and_scaling has curve_font defined by zero or one Curve_style_font objects. Each Curve_style_font defines curve_font for zero, one, or more Curve_style_font_and_scaling objects. This assertion is established through a SELECT type.

4.3.295 Cutout_edge_segment to Inter_stratum_feature_edge_segment_template

Each Cutout_edge_segment has usage_definition defined by exactly one Inter_stratum_feature_edge_segment_template. Each Inter_stratum_feature_edge_segment_template defines usage_definition for zero, one, or more Cutout_edge_segment objects.

4.3.296 Cutout_edge_segment to Edge_segment_vertex

Each Cutout_edge_segment has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Cutout_edge_segment objects. Each Cutout_edge_segment has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Cutout_edge_segment objects.

4.3.297 Cutout_edge_segment to Cutout

Each Cutout_edge_segment has composed_cutout defined by exactly one Cutout. Each Cutout defines composed_cutout for zero, one, or more Cutout_edge_segment objects.

4.3.298 Datum_axis_and_tolerance_zone_orienting_relationship to Datum_axis

Each Datum_axis_and_tolerance_zone_orienting_relationship has orienting_datum_axis defined by exactly one Datum_axis. Each Datum_axis defines orienting_datum_axis for zero, one, or more Datum_axis_and_tolerance_zone_orienting_relationship objects.

4.3.299 Datum_axis_and_tolerance_zone_orienting_relationship to Tolerance_zone

Each Datum_axis_and_tolerance_zone_orienting_relationship has oriented_tolerance_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines oriented_tolerance_zone for zero, one, or more Datum_axis_and_tolerance_zone_orienting_relationship objects.

4.3.300 Datum_axis_related_orientation to Datum_axis_and_tolerance_zone_orienting_relationship

Each Datum_axis_related_orientation has orienting_relationship defined by exactly one Datum_axis_and_tolerance_zone_orienting_relationship. Each Datum_axis_and_tolerance_zone_orienting_relationship defines orienting_relationship for zero, one, or more Datum_axis_related_orientation objects.

4.3.301 Datum_based_direction_vector to Datum_plane

Each Datum_based_direction_vector has vector_of defined by exactly one Datum_plane. Each Datum_plane defines vector_of for zero, one, or more Datum_based_direction_vector objects.

4.3.302 Datum_based_direction_vector to Axis_placement

Each Datum_based_direction_vector has direction_vector_representation defined by exactly one Axis_placement. Each Axis_placement defines direction_vector_representation for zero, one, or more Datum_based_direction_vector objects.

4.3.303 Datum_based_length_measure to Datum_plane

Each Datum_based_length_measure has of_datum defined by exactly one Datum_plane. Each Datum_plane defines of_datum for zero, one, or more Datum_based_length_measure objects.

4.3.304 Datum_based_length_measure to Orientation

Each Datum_based_length_measure has measure_orientation defined by zero or one Orientation objects. Each Orientation defines measure_orientation for zero, one, or more Datum_based_length_measure objects. This assertion is established through a SELECT type.

4.3.305 Datum_based_length_measure to Length_data_element

Each Datum_based_length_measure has measure defined by exactly one Length_data_element. Each Length_data_element defines measure for zero, one, or more Datum_based_length_measure objects.

4.3.306 Datum_based_vector_orientation to Direction_element

Each Datum_based_vector_orientation has direction_components defined by exactly two Direction_element objects. Each Direction_element defines direction_components for one or more Datum_based_vector_orientation objects.

4.3.307 Datum_difference_analytical_model_port_assignment to Datum_difference

Each Datum_difference_analytical_model_port_assignment has of_datum_difference defined by exactly one Datum_difference. Each Datum_difference defines of_datum_difference for zero, one, or more Datum_difference_analytical_model_port_assignment objects.

4.3.308 Datum_difference_analytical_model_port_assignment to Analytical_model_port

Each Datum_difference_analytical_model_port_assignment has assigned_analytical_model_port defined by exactly one Analytical_model_port. Each Analytical_model_port defines assigned_analytical_model_port for zero, one, or more Datum_difference_analytical_model_port_assignment objects.

4.3.309 Datum_difference_based_characteristic to Datum_difference

Each Datum_difference_based_characteristic has of_datum_difference defined by exactly one Datum_difference. Each Datum_difference defines of_datum_difference for zero, one, or more Datum_difference_based_characteristic objects.

4.3.310 Datum_difference_based_model_parameter to Datum_difference

Each Datum_difference_based_model_parameter has of_datum_difference defined by exactly one Datum_difference. Each Datum_difference defines of_datum_difference for zero, one, or more Datum_difference_based_model_parameter objects.

4.3.311 Datum_difference_functional_unit_usage_view_terminal_assignment to Datum_difference

Each Datum_difference_functional_unit_usage_view_terminal_assignment has assigned_datum_difference defined by exactly one Datum_difference. Each Datum_difference defines assigned_datum_difference for zero, one, or more Datum_difference_functional_unit_usage_view_terminal_assignment objects.

4.3.312 Datum_difference_functional_unit_usage_view_terminal_assignment to Functional_unit_usage_view_terminal_definition

Each Datum_difference_functional_unit_usage_view_terminal_assignment has assigned_functional_terminal_definition defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines assigned_functional_terminal_definition for zero, one, or more Datum_difference_functional_unit_usage_view_terminal_assignment objects.

4.3.313 Datum_feature_material_condition_property to Datum_feature_usage_in_datum_system_relationship

Each Datum_feature_material_condition_property has applied_to defined by exactly one Datum_feature_usage_in_datum_system_relationship. Each Datum_feature_usage_in_datum_system_relationship defines applied_to for zero, one, or more Datum_feature_material_condition_property objects.

4.3.314 Datum_feature_usage_in_common_datum_relationship to Common_datum

Each Datum_feature_usage_in_common_datum_relationship has relating_shape_element defined by exactly one Common_datum. Each Common_datum defines relating_shape_element for zero, one, or more Datum_feature_usage_in_common_datum_relationship objects.

4.3.315 Datum_feature_usage_in_datum_relationship to Datum_feature

Each Datum_feature_usage_in_datum_relationship has related_shape_element defined by exactly one Datum_feature. Each Datum_feature defines related_shape_element for one or more Datum_feature_usage_in_datum_relationship objects.

4.3.316 Datum_feature_usage_in_datum_relationship to Datum

Each Datum_feature_usage_in_datum_relationship has relating_shape_element defined by exactly one Datum. Each Datum defines relating_shape_element for zero, one, or more Datum_feature_usage_in_datum_relationship objects.

4.3.317 Datum_feature_usage_in_datum_system_relationship to Datum_feature

Each Datum_feature_usage_in_datum_system_relationship has used_datum_feature defined by exactly one Datum_feature. Each Datum_feature defines used_datum_feature for zero, one, or more Datum_feature_usage_in_datum_system_relationship objects.

4.3.318 Datum_feature_usage_in_datum_system_relationship to Datum_system

Each Datum_feature_usage_in_datum_system_relationship has established_datum_system defined by exactly one Datum_system. Each Datum_system defines established_datum_system for zero, one, or more Datum_feature_usage_in_datum_system_relationship objects.

4.3.319 Datum_feature_usage_in_single_datum_relationship to Single_datum

Each Datum_feature_usage_in_single_datum_relationship has relating_shape_element defined by exactly one Single_datum. Each Single_datum defines relating_shape_element for zero, one, or more Datum_feature_usage_in_single_datum_relationship objects.

4.3.320 Datum_point to Volume_shape_intersection

Each Datum_point has associated_volume_shape_intersection defined by exactly one Volume_shape_intersection. Each Volume_shape_intersection defines associated_volume_shape_intersection for zero, one, or more Datum_point objects.

4.3.321 Datum_point to Cartesian_point

Each Datum_point has location_in_3d defined by exactly one Cartesian_point. Each Cartesian_point defines location_in_3d for zero, one, or more Datum_point objects. Each Datum_point has location_in_2d defined by exactly one Cartesian_point. Each Cartesian_point defines location_in_2d for zero, one, or more Datum_point objects.

4.3.322 Datum_precedence_assignment to Datum_usage_in_datum_system_relationship

Each Datum_precedence_assignment has assigned_to defined by exactly one Datum_usage_in_datum_system_relationship. Each Datum_usage_in_datum_system_relationship defines assigned_to for zero, one, or more Datum_precedence_assignment objects.

4.3.323 Datum_reference_frame to Datum_point

Each Datum_reference_frame has reference_origin defined by exactly one Datum_point. Each Datum_point defines reference_origin for zero, one, or more Datum_reference_frame objects.

4.3.324 Datum_reference_frame to Perpendicular_shape_element_relationship

Each Datum_reference_frame has reference_axis_pairs defined by exactly three Perpendicular_shape_element_relationship objects. Each Perpendicular_shape_element_relationship defines reference_axis_pairs for zero, one, or more Datum_reference_frame objects. Each Datum_reference_frame has reference_plane_pairs defined by exactly three Perpendicular_shape_element_relationship objects. Each Perpendicular_shape_element_relationship defines reference_plane_pairs for zero, one, or more Datum_reference_frame objects.

4.3.325 Datum_reference_frame to Datum_axis

Each Datum_reference_frame has reference_axes defined by exactly three Datum_axis objects. Each Datum_axis defines reference_axes for zero, one, or more Datum_reference_frame objects.

4.3.326 Datum_reference_frame to Datum_plane

Each Datum_reference_frame has reference_planes defined by exactly three Datum_plane objects. Each Datum_plane defines reference_planes for zero, one, or more Datum_reference_frame objects.

4.3.327 Datum_system to Datum_reference_frame

Each Datum_system has established_datum_reference_frame defined by exactly one Datum_reference_frame. Each Datum_reference_frame defines established_datum_reference_frame for one or more Datum_system objects.

4.3.328 Datum_system_defining_conditions to Datum_precedence_assignment

Each Datum_system_defining_conditions has assigned_datum_precedences defined by one, two, or three Datum_precedence_assignment objects. Each Datum_precedence_assignment defines assigned_datum_precedences for zero, one, or more Datum_system_defining_conditions objects.

4.3.329 Datum_system_defining_conditions to Datum_system

Each Datum_system_defining_conditions has conditions_of defined by exactly one Datum_system. Each Datum_system defines conditions_of for zero, one, or more Datum_system_defining_conditions objects.

4.3.330 Datum_system_defining_conditions_with_material_conditions to Datum_feature_material_condition_property

Each Datum_system_defining_conditions_with_material_conditions has applied_material_condition_properties defined by one or more Datum_feature_material_condition_property objects. Each Datum_feature_material_condition_property defines applied_material_condition_properties for zero, one, or more Datum_system_defining_conditions_with_material_conditions objects.

4.3.331 Datum_target_usage_in_datum_target_set_relationship to Datum_target

Each Datum_target_usage_in_datum_target_set_relationship has used_datum_target defined by exactly one Datum_target. Each Datum_target defines used_datum_target for one or more Datum_target_usage_in_datum_target_set_relationship objects.

4.3.332 Datum_target_usage_in_datum_target_set_relationship to Datum_target_set

Each Datum_target_usage_in_datum_target_set_relationship has defined_datum_target_set defined by exactly one Datum_target_set. Each Datum_target_set defines defined_datum_target_set for one or more Datum_target_usage_in_datum_target_set_relationship objects.

4.3.333 Datum_usage_in_datum_system_relationship to Datum

Each Datum_usage_in_datum_system_relationship has used_datum defined by exactly one Datum. Each Datum defines used_datum for zero, one, or more Datum_usage_in_datum_system_relationship objects.

4.3.334 Datum_usage_in_datum_system_relationship to Datum_system

Each Datum_usage_in_datum_system_relationship has established_datum_system defined by exactly one Datum_system. Each Datum_system defines established_datum_system for one, two, or three Datum_usage_in_datum_system_relationship objects.

4.3.335 Default_assembly_bond_shape to Assembly_bond_definition

Each Default_assembly_bond_shape has characterized_bond_definition defined by exactly one Assembly_bond_definition. Each Assembly_bond_definition defines characterized_bond_definition for zero or one Default_assembly_bond_shape objects.

4.3.336 Default_attachment_size_based_land_physical_template to Connection_zone

Each Default_attachment_size_based_land_physical_template has maximum_attachment_region_size defined by exactly one Connection_zone. Each Connection_zone defines maximum_attachment_region_size for zero, one, or more Default_attachment_size_based_land_physical_template objects. Each Default_attachment_size_based_land_physical_template has minimum_attachment_region_size defined by exactly one Connection_zone. Each Connection_zone defines minimum_attachment_region_size for zero, one, or more Default_attachment_size_based_land_physical_template objects.

4.3.337 Default_component_termination_passage_based_land_physical_template to Default_component_termination_passage_definition

Each Default_component_termination_passage_based_land_physical_template has of_passage_technology defined by exactly one Default_component_termination_passage_definition. Each Default_component_termination_passage_definition defines of_passage_technology for zero, one, or more Default_component_termination_passage_based_land_physical_template objects.

4.3.338 Default_component_termination_passage_definition to Length_data_element

Each Default_component_termination_passage_definition has minimum_as_finished_passage_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_passage_extent for zero, one, or more Default_component_termination_passage_definition objects. Each Default_component_termination_passage_definition has minimum_as_finished_deposition_thickness defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_deposition_thickness for zero, one, or more Default_component_termination_passage_definition objects. Each Default_component_termination_passage_definition has maximum_allowed_component_terminal_extent defined by exactly one Length_data_element. Each Length_data_element defines maximum_allowed_component_terminal_extent for zero, one, or more Default_component_termination_passage_definition objects. Each Default_component_termination_passage_definition has minimum_allowed_component_terminal_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_allowed_component_terminal_extent for zero, one, or more Default_component_termination_passage_definition objects.

4.3.339 Default_component_termination_passage_definition to Ee_material

Each Default_component_termination_passage_definition has deposition_material defined by exactly one Ee_material. Each Ee_material defines deposition_material for zero, one, or more Default_component_termination_passage_definition objects.

4.3.340 Default_land_physical_template to Design_layer_technology

Each Default_land_physical_template has of_stratum_technology defined by exactly one Design_layer_technology. Each Design_layer_technology defines of_stratum_technology for zero, one, or more Default_land_physical_template objects.

4.3.341 Default_passage_based_land_physical_template to Length_data_element

Each Default_passage_based_land_physical_template has minimum_annular_ring defined by exactly one Length_data_element. Each Length_data_element defines minimum_annular_ring for zero, one, or more Default_passage_based_land_physical_template objects. Each Default_passage_based_land_physical_template has minimum_fabrication_allowance defined by exactly one Length_data_element. Each Length_data_element defines minimum_fabrication_allowance for zero, one, or more Default_passage_based_land_physical_template objects.

4.3.342 Default_passage_based_land_physical_template to Inter_stratum_extent

Each Default_passage_based_land_physical_template has of_stratum_extent defined by exactly one Inter_stratum_extent. Each Inter_stratum_extent defines of_stratum_extent for zero, one, or more Default_passage_based_land_physical_template objects.

4.3.343 Default_passage_based_land_physical_template to Passage_technology

Each Default_passage_based_land_physical_template has of_passage_technology defined by exactly one Passage_technology. Each Passage_technology defines of_passage_technology for zero, one, or more Default_passage_based_land_physical_template objects.

4.3.344 Default_trace_template to Stratum_technology

Each Default_trace_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Default_trace_template objects.

4.3.345 Default_unsupported_passage_based_land_physical_template to Default_unsupported_passage_definition

Each Default_unsupported_passage_based_land_physical_template has of_passage_technology defined by exactly one Default_unsupported_passage_definition. Each Default_unsupported_passage_definition defines of_passage_technology for zero, one, or more Default_unsupported_passage_based_land_physical_template objects.

4.3.346 Default_unsupported_passage_definition to Length_data_element

Each Default_unsupported_passage_definition has minimum_as_finished_passage_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_passage_extent for zero, one, or more Default_unsupported_passage_definition objects.

4.3.347 Default_via_based_land_physical_template to Default_via_definition

Each Default_via_based_land_physical_template has of_passage_technology defined by exactly one Default_via_definition. Each Default_via_definition defines of_passage_technology for zero, one, or more Default_via_based_land_physical_template objects.

4.3.348 Default_via_definition to Length_data_element

Each Default_via_definition has minimum_as_finished_passage_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_passage_extent for zero, one, or more Default_via_definition objects. Each Default_via_definition has minimum_as_finished_deposition_thickness defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_deposition_thickness for zero, one, or more Default_via_definition objects.

4.3.349 Default_via_definition to Ee_material

Each Default_via_definition has deposition_material defined by exactly one Ee_material. Each Ee_material defines deposition_material for zero, one, or more Default_via_definition objects.

4.3.350 Delete_design_object_management_relationship to Managed_design_object

Each Delete_design_object_management_relationship has previous_design_object defined by exactly one Managed_design_object. Each Managed_design_object defines previous_design_object for zero, one, or more Delete_design_object_management_relationship objects.

4.3.351 Dependently_located_layer_connection_point to Inter_stratum_feature

Each Dependently_located_layer_connection_point has associated_design_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines associated_design_object for zero, one, or more Dependently_located_layer_connection_point objects. This assertion is established through a SELECT type.

4.3.352 Dependently_located_layer_connection_point to Embedded_component_terminal

Each Dependently_located_layer_connection_point has associated_design_object defined by zero or one Embedded_component_terminal objects. Each Embedded_component_terminal defines associated_design_object for zero, one, or more Dependently_located_layer_connection_point objects. This assertion is established through a SELECT type.

4.3.353 Dependently_located_layer_connection_point to Interconnect_module_macro_component_join_terminal

Each Dependently_located_layer_connection_point has associated_design_object defined by zero or one Interconnect_module_macro_component_join_terminal objects. Each Interconnect_module_macro_component_join_terminal defines associated_design_object for zero, one, or more Dependently_located_layer_connection_point objects. This assertion is established through a SELECT type.

4.3.354 Dependently_located_layer_connection_point to Interconnect_module_terminal

Each Dependently_located_layer_connection_point has associated_design_object defined by zero or one Interconnect_module_terminal objects. Each Interconnect_module_terminal defines associated_design_object for zero, one, or more Dependently_located_layer_connection_point objects. This assertion is established through a SELECT type.

4.3.355 Dependently_located_layer_connection_point to Printed_component_join_terminal

Each Dependently_located_layer_connection_point has associated_design_object defined by zero or one Printed_component_join_terminal objects. Each Printed_component_join_terminal defines associated_design_object for zero, one, or more Dependently_located_layer_connection_point objects. This assertion is established through a SELECT type.

4.3.356 Dependent_material_removal_feature_template to Default_land_physical_template

Each `Dependent_material_removal_feature_template` has `associated_template` defined by exactly one `Default_land_physical_template`. Each `Default_land_physical_template` defines `associated_template` for zero, one, or more `Dependent_material_removal_feature_template` objects.

4.3.357 Design_composition_path to Functional_unit

Each `Design_composition_path` has `design_functional_unit` defined by exactly one `Functional_unit`. Each `Functional_unit` defines `design_functional_unit` for zero, one, or more `Design_composition_path` objects. Each `Design_composition_path` has `path_element` defined by one or more `Functional_unit` objects. Each `Functional_unit` defines `path_element` for zero, one, or more `Design_composition_path` objects.

4.3.358 Design_functional_unit_allocation_to_assembly_component to Coordinated_characteristic

Each `Design_functional_unit_allocation_to_assembly_component` has `of_property` defined by one or more `Coordinated_characteristic` objects. Each `Coordinated_characteristic` defines `of_property` for zero, one, or more `Design_functional_unit_allocation_to_assembly_component` objects.

4.3.359 Design_functional_unit_allocation_to_assembly_component to Assembly_component

Each `Design_functional_unit_allocation_to_assembly_component` has `implementation` defined by exactly one `Assembly_component`. Each `Assembly_component` defines `implementation` for zero, one, or more `Design_functional_unit_allocation_to_assembly_component` objects.

4.3.360 Design_functional_unit_allocation_to_assembly_component to Design_composition_path

Each `Design_functional_unit_allocation_to_assembly_component` has `functional_design_definition_path` defined by exactly one `Design_composition_path`. Each `Design_composition_path` defines `functional_design_definition_path` for zero, one, or more `Design_functional_unit_allocation_to_assembly_component` objects.

4.3.361 Design_functional_unit_allocation_to_assembly_joint to Assembly_joint

Each `Design_functional_unit_allocation_to_assembly_joint` has `implementation` defined by exactly one `Assembly_joint`. Each `Assembly_joint` defines `implementation` for zero, one, or more `Design_functional_unit_allocation_to_assembly_joint` objects.

4.3.362 Design_functional_unit_allocation_to_assembly_joint to Design_composition_path

Each Design_functional_unit_allocation_to_assembly_joint has functional_design_definition_path defined by exactly one Design_composition_path. Each Design_composition_path defines functional_design_definition_path for zero, one, or more Design_functional_unit_allocation_to_assembly_joint objects.

4.3.363 Design_functional_unit_allocation_to_fabrication_joint to Fabrication_joint

Each Design_functional_unit_allocation_to_fabrication_joint has implementation defined by exactly one Fabrication_joint. Each Fabrication_joint defines implementation for zero, one, or more Design_functional_unit_allocation_to_fabrication_joint objects.

4.3.364 Design_functional_unit_allocation_to_fabrication_joint to Design_composition_path

Each Design_functional_unit_allocation_to_fabrication_joint has functional_design_definition_path defined by exactly one Design_composition_path. Each Design_composition_path defines functional_design_definition_path for zero, one, or more Design_functional_unit_allocation_to_fabrication_joint objects.

4.3.365 Design_functional_unit_allocation_to_reference_functional_unit to Design_composition_path

Each Design_functional_unit_allocation_to_reference_functional_unit has design_definition_path defined by exactly one Design_composition_path. Each Design_composition_path defines design_definition_path for zero, one, or more Design_functional_unit_allocation_to_reference_functional_unit objects.

4.3.366 Design_functional_unit_allocation_to_reference_functional_unit to Reference_composition_path

Each Design_functional_unit_allocation_to_reference_functional_unit has reference_definition_path defined by exactly one Reference_composition_path. Each Reference_composition_path defines reference_definition_path for zero, one, or more Design_functional_unit_allocation_to_reference_functional_unit objects.

4.3.367 Design_intent_modification to Laminate_component

Each Design_intent_modification has modification_causal_item defined by zero or one Laminate_component objects. Each Laminate_component defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.368 Design_intent_modification to Inter_stratum_feature

Each Design_intent_modification has modification_causal_item defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.369 Design_intent_modification to Stratum_feature

Each Design_intent_modification has modification_causal_item defined by zero or one Stratum_feature objects. Each Stratum_feature defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type. Each Design_intent_modification has design_intent defined by exactly one Stratum_feature. Each Stratum_feature defines design_intent for zero, one, or more Design_intent_modification objects.

4.3.370 Design_intent_modification to Stratum

Each Design_intent_modification has modification_causal_item defined by zero or one Stratum objects. Each Stratum defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.371 Design_intent_modification to Stratum_surface

Each Design_intent_modification has modification_causal_item defined by zero or one Stratum_surface objects. Each Stratum_surface defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.372 Design_intent_modification to Ee_requirement_occurrence

Each Design_intent_modification has modification_causal_item defined by zero or one Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.373 Design_intent_modification to Part_feature

Each Design_intent_modification has modification_causal_item defined by zero or one Part_feature objects. Each Part_feature defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.374 Design_intent_modification to Assembly_component

Each Design_intent_modification has modification_causal_item defined by zero or one Assembly_component objects. Each Assembly_component defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.375 Design_intent_modification to Component_external_reference

Each Design_intent_modification has modification_causal_item defined by zero or one Component_external_reference objects. Each Component_external_reference defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.376 Design_intent_modification to Component_feature

Each Design_intent_modification has modification_causal_item defined by zero or one Component_feature objects. Each Component_feature defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.377 Design_intent_modification to Component_feature_external_reference

Each Design_intent_modification has modification_causal_item defined by zero or one Component_feature_external_reference objects. Each Component_feature_external_reference defines modification_causal_item for zero, one, or more Design_intent_modification objects. This assertion is established through a SELECT type.

4.3.378 Design_intent_modification_3d_position to Design_intent_modification_non_planar_2d_shape

Each Design_intent_modification_3d_position has feature_shape defined by zero or one Design_intent_modification_non_planar_2d_shape objects. Each Design_intent_modification_non_planar_2d_shape defines feature_shape for zero, one, or more Design_intent_modification_3d_position objects. This assertion is established through a SELECT type.

4.3.379 Design_intent_modification_3d_position to Design_intent_modification_3d_shape

Each Design_intent_modification_3d_position has feature_shape defined by zero or one Design_intent_modification_3d_shape objects. Each Design_intent_modification_3d_shape defines feature_shape for zero, one, or more Design_intent_modification_3d_position objects. This assertion is established through a SELECT type.

4.3.380 Design_intent_modification_3d_position to Axis_placement_3d

Each Design_intent_modification_3d_position has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Design_intent_modification_3d_position objects.

4.3.381 Design_intent_modification_3d_position to Physical_unit_3d_shape

Each Design_intent_modification_3d_position has design_shape defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines design_shape for zero, one, or more Design_intent_modification_3d_position objects.

4.3.382 Design_intent_modification_3d_shape to Design_intent_modification

Each Design_intent_modification_3d_shape has represented_design_intent_modification defined by exactly one Design_intent_modification. Each Design_intent_modification defines represented_design_intent_modification for zero, one, or more Design_intent_modification_3d_shape objects.

4.3.383 Design_intent_modification_non_planar_2d_shape to Design_intent_modification

Each Design_intent_modification_non_planar_2d_shape has represented_design_intent_modification defined by exactly one Design_intent_modification. Each Design_intent_modification defines represented_design_intent_modification for zero, one, or more Design_intent_modification_non_planar_2d_shape objects.

4.3.384 Design_intent_modification_planar_position to Design_intent_modification_styled_area_representation

Each Design_intent_modification_planar_position has feature_shape defined by zero or one Design_intent_modification_styled_area_representation objects. Each Design_intent_modification_styled_area_representation defines feature_shape for zero, one, or more Design_intent_modification_planar_position objects. This assertion is established through a SELECT type.

4.3.385 Design_intent_modification_planar_position to Design_intent_modification_planar_shape

Each Design_intent_modification_planar_position has feature_shape defined by zero or one Design_intent_modification_planar_shape objects. Each Design_intent_modification_planar_shape defines feature_shape for zero, one, or more Design_intent_modification_planar_position objects. This assertion is established through a SELECT type.

4.3.386 Design_intent_modification_planar_position to Axis_placement_2d

Each Design_intent_modification_planar_position has transformation defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines transformation for zero, one, or more Design_intent_modification_planar_position objects.

4.3.387 Design_intent_modification_planar_position to Physical_unit_planar_shape

Each Design_intent_modification_planar_position has design_shape defined by exactly one Physical_unit_planar_shape. Each Physical_unit_planar_shape defines design_shape for zero, one, or more Design_intent_modification_planar_position objects.

4.3.388 Design_intent_modification_planar_shape to Design_intent_modification

Each Design_intent_modification_planar_shape has represented_design_intent_modification defined by exactly one Design_intent_modification. Each Design_intent_modification defines represented_design_intent_modification for zero, one, or more Design_intent_modification_planar_shape objects.

4.3.389 Design_intent_modification_sequence to Design_intent_modification

Each Design_intent_modification_sequence has subsequent_modification defined by zero or one Design_intent_modification objects. Each Design_intent_modification defines subsequent_modification for zero, one, or more Design_intent_modification_sequence objects. This assertion is established through a SELECT type. Each Design_intent_modification_sequence has precedent_modification defined by zero or one Design_intent_modification objects. Each Design_intent_modification defines precedent_modification for zero, one, or more Design_intent_modification_sequence objects. This assertion is established through a SELECT type.

4.3.390 Design_intent_modification_sequence to Material_removal_laminate_component

Each Design_intent_modification_sequence has subsequent_modification defined by zero or one Material_removal_laminate_component objects. Each Material_removal_laminate_component defines subsequent_modification for zero, one, or more Design_intent_modification_sequence objects. This assertion is established through a SELECT type. Each Design_intent_modification_sequence has precedent_modification defined by zero or one Material_removal_laminate_component objects. Each Material_removal_laminate_component defines precedent_modification for zero, one, or more Design_intent_modification_sequence objects. This assertion is established through a SELECT type.

4.3.391 Design_intent_modification_styled_area_representation to Fill_area_occurrence

Each Design_intent_modification_styled_area_representation has shape_styled_area_representation defined by zero or one Fill_area_occurrence objects. Each Fill_area_occurrence defines shape_styled_area_representation for zero, one, or more Design_intent_modification_styled_area_representation objects. This assertion is established through a SELECT type.

4.3.392 Design_intent_modification_styled_area_representation to Styled_curve_occurrence

Each Design_intent_modification_styled_area_representation has shape_styled_area_representation defined by zero or one Styled_curve_occurrence objects. Each Styled_curve_occurrence defines shape_styled_area_representation for zero, one, or more Design_intent_modification_styled_area_representation objects. This assertion is established through a SELECT type.

4.3.393 Design_intent_modification_styled_area_representation to Design_intent_modification

Each Design_intent_modification_styled_area_representation has represented_design_intent_modification defined by exactly one Design_intent_modification. Each Design_intent_modification defines represented_design_intent_modification for zero, one, or more Design_intent_modification_styled_area_representation objects.

4.3.394 Design_intent_modification_styled_area_representation to Design_intent_modification_planar_shape

Each Design_intent_modification_styled_area_representation has derived_planar_shape defined by one or more Design_intent_modification_planar_shape objects. Each Design_intent_modification_planar_shape defines derived_planar_shape for zero, one, or more Design_intent_modification_styled_area_representation objects.

4.3.395 Design_layer_stratum to Design_layer_technology

Each Design_layer_stratum has of_technology defined by exactly one Design_layer_technology. Each Design_layer_technology defines of_technology for zero, one, or more Design_layer_stratum objects.

4.3.396 Design_layer_technology to Length_data_element

Each Design_layer_technology has minimum_finished_feature_spacing defined by exactly one Length_data_element. Each Length_data_element defines minimum_finished_feature_spacing for zero, one, or more Design_layer_technology objects. Each Design_layer_technology has minimum_finished_feature_size defined by exactly one Length_data_element. Each Length_data_element defines minimum_finished_feature_size for zero, one, or more Design_layer_technology objects.

4.3.397 Design_object_management_relationship to Ee_product_definition

Each Design_object_management_relationship has current_design defined by exactly one Ee_product_definition. Each Ee_product_definition defines current_design for zero, one, or more Design_object_management_relationship objects. Each Design_object_management_relationship has previous_design defined by exactly one Ee_product_definition. Each Ee_product_definition defines previous_design for zero, one, or more Design_object_management_relationship objects.

4.3.398 Design_view_connection_zone_terminal_shape_relationship to Conductive_interconnect_element_terminal

Each Design_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Conductive_interconnect_element_terminal objects. Each Conductive_interconnect_element_terminal defines associating_terminal for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.399 Design_view_connection_zone_terminal_shape_relationship to Interconnect_component_interface_terminal

Each Design_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Interconnect_component_interface_terminal objects. Each Interconnect_component_interface_terminal defines associating_terminal for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.400 Design_view_connection_zone_terminal_shape_relationship to Interconnect_component_join_terminal

Each Design_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Interconnect_component_join_terminal objects. Each Interconnect_component_join_terminal defines associating_terminal for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.401 Design_view_connection_zone_terminal_shape_relationship to Axis_placement

Each Design_view_connection_zone_terminal_shape_relationship has associated_usage_placement defined by exactly one Axis_placement. Each Axis_placement defines associated_usage_placement for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects.

4.3.402 Design_view_connection_zone_terminal_shape_relationship to Feature_shape_definition

Each Design_view_connection_zone_terminal_shape_relationship has associated_connection_zone_shape_definition defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associated_connection_zone_shape_definition for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects. Each Design_view_connection_zone_terminal_shape_relationship has associating_terminal_shape defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associating_terminal_shape for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects.

4.3.403 Design_view_connection_zone_terminal_shape_relationship to Connection_zone

Each Design_view_connection_zone_terminal_shape_relationship has associated_usage defined by exactly one Connection_zone. Each Connection_zone defines associated_usage for zero, one, or more Design_view_connection_zone_terminal_shape_relationship objects.

4.3.404 Design_view_terminal_component_shape_relationship to Conductive_interconnect_element

Each Design_view_terminal_component_shape_relationship has associating_component defined by zero or one Conductive_interconnect_element objects. Each Conductive_interconnect_element defines associating_component for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.405 Design_view_terminal_component_shape_relationship to Packaged_component

Each Design_view_terminal_component_shape_relationship has associating_component defined by zero or one Packaged_component objects. Each Packaged_component defines associating_component for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.406 Design_view_terminal_component_shape_relationship to Physical_component

Each Design_view_terminal_component_shape_relationship has associating_component defined by zero or one Physical_component objects. Each Physical_component defines associating_component for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.407 Design_view_terminal_component_shape_relationship to Axis_placement

Each Design_view_terminal_component_shape_relationship has associated_usage_placement defined by exactly one Axis_placement. Each Axis_placement defines associated_usage_placement for zero, one, or more Design_view_terminal_component_shape_relationship objects.

4.3.408 Design_view_terminal_component_shape_relationship to Feature_shape_definition

Each Design_view_terminal_component_shape_relationship has associated_terminal_shape_definition defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associated_terminal_shape_definition for zero, one, or more Design_view_terminal_component_shape_relationship objects. Each Design_view_terminal_component_shape_relationship has associating_component_shape defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associating_component_shape for zero, one, or more Design_view_terminal_component_shape_relationship objects.

4.3.409 Design_view_terminal_component_shape_relationship to Conductive_interconnect_element_terminal

Each Design_view_terminal_component_shape_relationship has associated_usage defined by zero or one Conductive_interconnect_element_terminal objects. Each Conductive_interconnect_element_terminal defines associated_usage for one or two Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.410 Design_view_terminal_component_shape_relationship to Interconnect_component_interface_terminal

Each Design_view_terminal_component_shape_relationship has associated_usage defined by zero or one Interconnect_component_interface_terminal objects. Each Interconnect_component_interface_terminal defines associated_usage for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.411 Design_view_terminal_component_shape_relationship to Interconnect_component_join_terminal

Each Design_view_terminal_component_shape_relationship has associated_usage defined by zero or one Interconnect_component_join_terminal objects. Each Interconnect_component_join_terminal defines associated_usage for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.412 Design_view_terminal_component_shape_relationship to Movable_packaged_component_join_terminal

Each Design_view_terminal_component_shape_relationship has associated_usage defined by zero or one Movable_packaged_component_join_terminal objects. Each Movable_packaged_component_join_terminal defines associated_usage for zero, one, or more Design_view_terminal_component_shape_relationship objects. This assertion is established through a SELECT type.

4.3.413 Device_2d_position to Physical_unit_planar_shape

Each Device_2d_position has device_shape defined by exactly one Physical_unit_planar_shape. Each Physical_unit_planar_shape defines device_shape for zero, one, or more Device_2d_position objects. Each Device_2d_position has part_shape defined by exactly one Physical_unit_planar_shape. Each Physical_unit_planar_shape defines part_shape for zero, one, or more Device_2d_position objects.

4.3.414 Device_2d_position to Axis_placement_2d

Each Device_2d_position has transformation defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines transformation for zero, one, or more Device_2d_position objects.

4.3.415 Device_2d_position to Part_device

Each Device_2d_position has placed_device defined by exactly one Part_device. Each Part_device defines placed_device for zero or one Device_2d_position objects.

4.3.416 Device_3d_position to Physical_unit_3d_shape

Each Device_3d_position has device_shape defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines device_shape for zero, one, or more Device_3d_position objects. Each Device_3d_position has part_shape defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines part_shape for zero, one, or more Device_3d_position objects.

4.3.417 Device_3d_position to Axis_placement_3d

Each Device_3d_position has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Device_3d_position objects.

4.3.418 Device_3d_position to Part_device

Each Device_3d_position has placed_device defined by exactly one Part_device. Each Part_device defines placed_device for zero or one Device_3d_position objects.

4.3.419 Device_terminal_map to Part_device_terminal

Each Device_terminal_map has mapped_device_terminal defined by exactly one Part_device_terminal. Each Part_device_terminal defines mapped_device_terminal for zero, one, or more Device_terminal_map objects.

4.3.420 Device_terminal_map to Packaged_part_terminal

Each Device_terminal_map has target_terminal defined by exactly one Packaged_part_terminal. Each Packaged_part_terminal defines target_terminal for zero, one, or more Device_terminal_map objects.

4.3.421 Device_terminal_map_relationship to Device_terminal_map

Each Device_terminal_map_relationship has relating_device_terminal_map defined by exactly one Device_terminal_map. Each Device_terminal_map defines relating_device_terminal_map for zero, one, or more Device_terminal_map_relationship objects. Each Device_terminal_map_relationship has related_device_terminal_map defined by exactly one Device_terminal_map. Each Device_terminal_map defines related_device_terminal_map for zero, one, or more Device_terminal_map_relationship objects.

4.3.422 Diametrical_boundary_size_characteristic to Tolerance_zone_circular_or_cylindrical_or_spherical_boundary

Each Diametrical_boundary_size_characteristic has sized_boundary defined by exactly one Tolerance_zone_circular_or_cylindrical_or_spherical_boundary. Each Tolerance_zone_circular_or_cylindrical_or_spherical_boundary defines sized_boundary for zero, one, or more Diametrical_boundary_size_characteristic objects.

4.3.423 Dielectric_crossover_area to Stratum_feature

Each Dielectric_crossover_area has resulting_dielectric_crossover defined by exactly one Stratum_feature. Each Stratum_feature defines resulting_dielectric_crossover for zero, one, or more Dielectric_crossover_area objects. Each Dielectric_crossover_area has relating_concept defined by exactly one Stratum_feature. Each Stratum_feature defines relating_concept for zero, one, or more Dielectric_crossover_area objects. Each Dielectric_crossover_area has related_concept defined by exactly one Stratum_feature. Each Stratum_feature defines related_concept for zero, one, or more Dielectric_crossover_area objects.

4.3.424 Dielectric_material_passage to Cutout

Each Dielectric_material_passage has precedent_passage defined by exactly one Cutout. Each Cutout defines precedent_passage for zero, one, or more Dielectric_material_passage objects.

4.3.425 Dielectric_material_passage to Ee_requirement_occurrence

Each Dielectric_material_passage has feature_material defined by zero or one Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines feature_material for zero, one, or more Dielectric_material_passage objects. This assertion is established through a SELECT type.

4.3.426 Dielectric_material_passage to Ee_material

Each Dielectric_material_passage has feature_material defined by zero or one Ee_material objects. Each Ee_material defines feature_material for zero, one, or more Dielectric_material_passage objects. This assertion is established through a SELECT type.

4.3.427 Dimensional_characteristic_representation to Dimensional_size

Each Dimensional_characteristic_representation has dimension defined by zero or one Dimensional_size objects. Each Dimensional_size defines dimension for zero, one, or more Dimensional_characteristic_representation objects. This assertion is established through a SELECT type.

4.3.428 Dimensional_characteristic_representation to Dimensional_location

Each Dimensional_characteristic_representation has dimension defined by zero or one Dimensional_location objects. Each Dimensional_location defines dimension for zero, one, or more Dimensional_characteristic_representation objects. This assertion is established through a SELECT type.

4.3.429 Dimensional_characteristic_representation to Shape_dimension_representation

Each Dimensional_characteristic_representation has representation defined by exactly one Shape_dimension_representation. Each Shape_dimension_representation defines representation for zero, one, or more Dimensional_characteristic_representation objects.

4.3.430 Dimensional_location to Shape_element_locating_relationship

Each Dimensional_location has located_elements defined by exactly one Shape_element_locating_relationship. Each Shape_element_locating_relationship defines located_elements for zero, one, or more Dimensional_location objects.

4.3.431 Dimensional_size to Shape_element

Each Dimensional_size has size_of defined by exactly one Shape_element. Each Shape_element defines size_of for zero, one, or more Dimensional_size objects.

4.3.432 Dimensional_size_based_on_opposing_boundaries to Composite_unit_shape_element

Each Dimensional_size_based_on_opposing_boundaries has size_of defined by exactly one Composite_unit_shape_element. Each Composite_unit_shape_element defines size_of for zero, one, or more Dimensional_size_based_on_opposing_boundaries objects.

4.3.433 Dimensional_size_based_on_opposing_boundaries to Shape_element_locating_relationship

Each Dimensional_size_based_on_opposing_boundaries has boundary_separation defined by exactly one Shape_element_locating_relationship. Each Shape_element_locating_relationship defines boundary_separation for zero, one, or more Dimensional_size_based_on_opposing_boundaries objects.

4.3.434 Dimension_along_curve to Shape_element

Each Dimension_along_curve has path defined by exactly one Shape_element. Each Shape_element defines path for zero, one, or more Dimension_along_curve objects.

4.3.435 Dimension_location_with_direction_vector to Axis_placement

Each Dimension_location_with_direction_vector has measurement_direction defined by zero or one Axis_placement objects. Each Axis_placement defines measurement_direction for zero, one, or more Dimension_location_with_direction_vector objects. This assertion is established through a SELECT type.

4.3.436 Dimension_location_with_direction_vector to Datum_based_vector_orientation

Each Dimension_location_with_direction_vector has measurement_direction defined by zero or one Datum_based_vector_orientation objects. Each Datum_based_vector_orientation defines measurement_direction for zero, one, or more Dimension_location_with_direction_vector objects. This assertion is established through a SELECT type.

4.3.437 Dimension_location_with_specified_datum_system to Datum_system_defining_conditions_without_material_conditions

Each Dimension_location_with_specified_datum_system has referenced_datum_system_defining_conditions defined by exactly one Datum_system_defining_conditions_without_material_conditions. Each Datum_system_defining_conditions_without_material_conditions defines referenced_datum_system_defining_conditions for zero, one, or more Dimension_location_with_specified_datum_system objects.

4.3.438 Dimension_qualifier to Dimensional_characteristic

Each Dimension_qualifier has qualified_dimensions defined by one or more Dimensional_characteristic objects. Each Dimensional_characteristic defines qualified_dimensions for zero, one, or more Dimension_qualifier objects.

4.3.439 Dimension_related_positional_boundary to Physical_feature_or_part_template

Each Dimension_related_positional_boundary has constrained_physical_feature defined by exactly one Physical_feature_or_part_template. Each Physical_feature_or_part_template defines constrained_physical_feature for zero, one, or more Dimension_related_positional_boundary objects.

4.3.440 Dimension_related_positional_boundary to Positional_boundary_member

Each Dimension_related_positional_boundary has boundary_members defined by one or more Positional_boundary_member objects. Each Positional_boundary_member defines boundary_members for zero, one, or more Dimension_related_positional_boundary objects.

4.3.441 Dimension_related_positional_boundary_definition to Positional_boundary_member_definition

Each Dimension_related_positional_boundary_definition has characterized_boundary_member_definitions defined by one or more Positional_boundary_member_definition objects. Each Positional_boundary_member_definition defines characterized_boundary_member_definitions for zero, one, or more Dimension_related_positional_boundary_definition objects.

4.3.442 Dimension_related_positional_boundary_definition to Dimension_related_positional_boundary

Each Dimension_related_positional_boundary_definition has defined_positional_boundary defined by exactly one Dimension_related_positional_boundary. Each Dimension_related_positional_boundary defines defined_positional_boundary for zero, one, or more Dimension_related_positional_boundary_definition objects.

4.3.443 Dimension_set to Dimensional_characteristic

Each Dimension_set has associated_dimensions defined by one or more Dimensional_characteristic objects. Each Dimensional_characteristic defines associated_dimensions for zero, one, or more Dimension_set objects.

4.3.444 Directed_axis to Shape_element

Each Directed_axis has end_shape_element defined by exactly one Shape_element. Each Shape_element defines end_shape_element for zero, one, or more Directed_axis objects. Each Directed_axis has start_shape_element defined by exactly one Shape_element. Each Shape_element defines start_shape_element for zero, one, or more Directed_axis objects.

4.3.445 Directed_axis to Centre_axis

Each Directed_axis has axis_to_be_oriented defined by exactly one Centre_axis. Each Centre_axis defines axis_to_be_oriented for zero, one, or more Directed_axis objects.

4.3.446 Direction_element to Datum_based_direction_vector

Each Direction_element has direction_of_half_datum_plane defined by exactly one Datum_based_direction_vector. Each Datum_based_direction_vector defines direction_of_half_datum_plane for zero, one, or more Direction_element objects. Each Direction_element has direction defined by exactly one Datum_based_direction_vector. Each Datum_based_direction_vector defines direction for zero, one, or more Direction_element objects.

4.3.447 Direction_element to Datum_plane

Each Direction_element has base_datum_plane defined by exactly one Datum_plane. Each Datum_plane defines base_datum_plane for zero, one, or more Direction_element objects.

4.3.448 Direction_element to Angle_data_element

Each Direction_element has angle_value defined by exactly one Angle_data_element. Each Angle_data_element defines angle_value for zero, one, or more Direction_element objects.

4.3.449 Documentation_layer_stratum to Documentation_layer_technology

Each Documentation_layer_stratum has of_technology defined by exactly one Documentation_layer_technology. Each Documentation_layer_technology defines of_technology for zero, one, or more Documentation_layer_stratum objects.

4.3.450 Edge_segment_cross_section to Part_template_3d_shape

Each Edge_segment_cross_section has projected_volume defined by exactly one Part_template_3d_shape. Each Part_template_3d_shape defines projected_volume for zero, one, or more Edge_segment_cross_section objects.

4.3.451 Edge_segment_cross_section to Edge_shape

Each Edge_segment_cross_section has surface_intersection_result defined by exactly one Edge_shape. Each Edge_shape defines surface_intersection_result for zero, one, or more Edge_segment_cross_section objects.

4.3.452 Ee_approval to Date_and_time

Each Ee_approval has signoff_date defined by zero or one Date_and_time objects. Each Date_and_time defines signoff_date for zero, one, or more Ee_approval objects. This assertion is established through a SELECT type.

4.3.453 Ee_approval to Date

Each Ee_approval has signoff_date defined by zero or one Date objects. Each Date defines signoff_date for zero, one, or more Ee_approval objects. This assertion is established through a SELECT type.

4.3.454 Ee_approval to Person_and_organization

Each Ee_approval has authorizer defined by zero or one Person_and_organization objects. Each Person_and_organization defines authorizer for zero, one, or more Ee_approval objects. This assertion is established through a SELECT type.

4.3.455 Ee_approval to Organization

Each Ee_approval has authorizer defined by zero or one Organization objects. Each Organization defines authorizer for zero, one, or more Ee_approval objects. This assertion is established through a SELECT type.

4.3.456 Ee_approval to Ee_text

Each Ee_approval has purpose defined by exactly one Ee_text. Each Ee_text defines purpose for zero, one, or more Ee_approval objects.

4.3.457 Ee_document to Person

Each Ee_document has author defined by one or more Person objects. Each Person defines author for zero, one, or more Ee_document objects.

4.3.458 Ee_document to Structured_text

Each Ee_document has text_usage defined by zero or one Structured_text objects. Each Structured_text defines text_usage for zero, one, or more Ee_document objects. This assertion is established through a SELECT type. Each Ee_document has key_phrase defined by one or more Structured_text objects. Each Structured_text defines key_phrase for zero, one, or more Ee_document objects.

4.3.459 Ee_document to Ee_text

Each Ee_document has text_usage defined by zero or one Ee_text objects. Each Ee_text defines text_usage for zero, one, or more Ee_document objects. This assertion is established through a SELECT type.

4.3.460 Ee_document to Security_classification

Each Ee_document has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Ee_document objects.

4.3.461 Ee_document to Date_and_time

Each Ee_document has creation_date defined by exactly one Date_and_time. Each Date_and_time defines creation_date for zero, one, or more Ee_document objects. Each Ee_document has change_date defined by one or more Date_and_time objects. Each Date_and_time defines change_date for zero, one, or more Ee_document objects.

4.3.462 Ee_document to Ee_document_identifier

Each Ee_document has base_identifier defined by exactly one Ee_document_identifier. Each Ee_document_identifier defines base_identifier for zero, one, or more Ee_document objects.

4.3.463 Ee_document to Ee_approval

Each Ee_document has document_approval defined by exactly one Ee_approval. Each Ee_approval defines document_approval for zero, one, or more Ee_document objects.

4.3.464 Ee_document to Person_and_organization

Each Ee_document has publisher defined by zero or one Person_and_organization objects. Each Person_and_organization defines publisher for zero, one, or more Ee_document objects. This assertion is established through a SELECT type.

4.3.465 Ee_document to Organization

Each Ee_document has publisher defined by zero or one Organization objects. Each Organization defines publisher for zero, one, or more Ee_document objects. This assertion is established through a SELECT type.

4.3.466 Ee_material to Ee_document

Each Ee_material has registration_document defined by exactly one Ee_document. Each Ee_document defines registration_document for zero, one, or more Ee_material objects.

4.3.467 Ee_product to Person_and_organization

Each Ee_product has design_owner defined by zero or one Person_and_organization objects. Each Person_and_organization defines design_owner for zero, one, or more Ee_product objects. This assertion is established through a SELECT type.

4.3.468 Ee_product to Organization

Each Ee_product has design_owner defined by zero or one Organization objects. Each Organization defines design_owner for zero, one, or more Ee_product objects. This assertion is established through a SELECT type.

4.3.469 Ee_product to Ee_text

Each Ee_product has description defined by exactly one Ee_text. Each Ee_text defines description for zero, one, or more Ee_product objects.

4.3.470 Ee_product_category to Ee_document

Each Ee_product_category has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Ee_product_category objects.

4.3.471 Ee_product_category to Parameter_assignment

Each Ee_product_category has category_specific_values defined by one or more Parameter_assignment objects. Each Parameter_assignment defines category_specific_values for zero, one, or more Ee_product_category objects.

4.3.472 Ee_product_category to Model_parameter

Each Ee_product_category has category_parameter defined by one or more Model_parameter objects. Each Model_parameter defines category_parameter for zero, one, or more Ee_product_category objects.

4.3.473 Ee_product_category_relationship to Ee_product_category

Each Ee_product_category_relationship has category defined by exactly one Ee_product_category. Each Ee_product_category defines category for zero, one, or more Ee_product_category_relationship objects. Each Ee_product_category_relationship has sub_category defined by exactly one Ee_product_category. Each Ee_product_category defines sub_category for zero, one, or more Ee_product_category_relationship objects.

4.3.474 Ee_product_configuration to Ee_product_model

Each Ee_product_configuration has model_of_product defined by exactly one Ee_product_model. Each Ee_product_model defines model_of_product for zero, one, or more Ee_product_configuration objects.

4.3.475 Ee_product_configuration to Ee_approval

Each Ee_product_configuration has product_configuration_approval defined by exactly one Ee_approval. Each Ee_approval defines product_configuration_approval for zero, one, or more Ee_product_configuration objects.

4.3.476 Ee_product_configuration to Person_and_organization

Each Ee_product_configuration has configuration_manager defined by zero or one Person_and_organization objects. Each Person_and_organization defines configuration_manager for zero, one, or more Ee_product_configuration objects. This assertion is established through a SELECT type.

4.3.477 Ee_product_configuration to Organization

Each Ee_product_configuration has configuration_manager defined by zero or one Organization objects. Each Organization defines configuration_manager for zero, one, or more Ee_product_configuration objects. This assertion is established through a SELECT type.

4.3.478 Ee_product_configuration to Ee_product_version

Each Ee_product_configuration has satisfying_product defined by one or more Ee_product_version objects. Each Ee_product_version defines satisfying_product for zero, one, or more Ee_product_configuration objects.

4.3.479 Ee_product_definition to Ee_document

Each Ee_product_definition has documentation defined by one or more Ee_document objects. Each Ee_document defines documentation for zero, one, or more Ee_product_definition objects.

4.3.480 Ee_product_definition to Date_and_time

Each Ee_product_definition has creation_date defined by exactly one Date_and_time. Each Date_and_time defines creation_date for zero, one, or more Ee_product_definition objects.

4.3.481 Ee_product_definition to Ee_approval

Each Ee_product_definition has product_definition_approval defined by exactly one Ee_approval. Each Ee_approval defines product_definition_approval for zero, one, or more Ee_product_definition objects.

4.3.482 Ee_product_definition to Ee_product_version

Each Ee_product_definition has version defined by exactly one Ee_product_version. Each Ee_product_version defines version for one or more Ee_product_definition objects.

4.3.483 Ee_product_definition to Ee_text

Each Ee_product_definition has context_description defined by exactly one Ee_text. Each Ee_text defines context_description for zero, one, or more Ee_product_definition objects.

4.3.484 Ee_product_definition to Security_classification

Each Ee_product_definition has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Ee_product_definition objects.

4.3.485 Ee_product_definition to Person_and_organization

Each Ee_product_definition has creator defined by zero or one Person_and_organization objects. Each Person_and_organization defines creator for zero, one, or more Ee_product_definition objects. This assertion is established through a SELECT type.

4.3.486 Ee_product_definition to Organization

Each Ee_product_definition has creator defined by zero or one Organization objects. Each Organization defines creator for zero, one, or more Ee_product_definition objects. This assertion is established through a SELECT type.

4.3.487 Ee_product_definition_based_constraint_occurrence to Ee_product_definition

Each Ee_product_definition_based_constraint_occurrence has constraining_part defined by exactly one Ee_product_definition. Each Ee_product_definition defines constraining_part for zero, one, or more Ee_product_definition_based_constraint_occurrence objects.

4.3.488 Ee_product_model to Ee_text

Each Ee_product_model has model_name defined by exactly one Ee_text. Each Ee_text defines model_name for zero, one, or more Ee_product_model objects.

4.3.489 Ee_product_related_product_category to Ee_product

Each Ee_product_related_product_category has of_product defined by one or more Ee_product objects. Each Ee_product defines of_product for one or more Ee_product_related_product_category objects.

4.3.490 Ee_product_specific_parameter_value_assignment to Parameter_assignment

Each Ee_product_specific_parameter_value_assignment has assigned_parameter defined by exactly one Parameter_assignment. Each Parameter_assignment defines assigned_parameter for zero, one, or more Ee_product_specific_parameter_value_assignment objects.

4.3.491 Ee_product_specific_parameter_value_assignment to Ee_product

Each Ee_product_specific_parameter_value_assignment has of_product defined by exactly one Ee_product. Each Ee_product defines of_product for zero, one, or more Ee_product_specific_parameter_value_assignment objects.

4.3.492 Ee_product_specific_parameter_value_assignment to Ee_product_related_product_category

Each Ee_product_specific_parameter_value_assignment has associated_product_category defined by exactly one Ee_product_related_product_category. Each Ee_product_related_product_category defines associated_product_category for zero, one, or more Ee_product_specific_parameter_value_assignment objects.

4.3.493 Ee_product_version to Person_and_organization

Each Ee_product_version has design_supplier defined by zero or one Person_and_organization objects. Each Person_and_organization defines design_supplier for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type. Each Ee_product_version has product_supplier defined by zero or one Person_and_organization objects. Each Person_and_organization defines product_supplier for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type. Each Ee_product_version has creator defined by zero or one Person_and_organization objects. Each Person_and_organization defines creator for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type.

4.3.494 Ee_product_version to Organization

Each Ee_product_version has design_supplier defined by zero or one Organization objects. Each Organization defines design_supplier for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type. Each Ee_product_version has product_supplier defined by zero or one Organization objects. Each Organization defines product_supplier for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type. Each Ee_product_version has creator defined by zero or one Organization objects. Each Organization defines creator for zero, one, or more Ee_product_version objects. This assertion is established through a SELECT type.

4.3.495 Ee_product_version to Ee_approval

Each Ee_product_version has product_version_approval defined by exactly one Ee_approval. Each Ee_approval defines product_version_approval for zero, one, or more Ee_product_version objects.

4.3.496 Ee_product_version to Security_classification

Each Ee_product_version has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Ee_product_version objects.

4.3.497 Ee_product_version to Ee_product

Each Ee_product_version has versioned_item defined by exactly one Ee_product. Each Ee_product defines versioned_item for one or more Ee_product_version objects.

4.3.498 Ee_product_version to Contract

Each Ee_product_version has authorizing_agreement defined by exactly one Contract. Each Contract defines authorizing_agreement for zero, one, or more Ee_product_version objects.

4.3.499 Ee_product_version_supersedence to Ee_product_version

Each Ee_product_version_supersedence has superseding_product_version defined by exactly one Ee_product_version. Each Ee_product_version defines superseding_product_version for zero, one, or more Ee_product_version_supersedence objects. Each Ee_product_version_supersedence has superseded_product_version defined by exactly one Ee_product_version. Each Ee_product_version defines superseded_product_version for zero, one, or more Ee_product_version_supersedence objects.

4.3.500 Ee_requirement_occurrence to Ee_text

Each Ee_requirement_occurrence has description defined by exactly one Ee_text. Each Ee_text defines description for zero, one, or more Ee_requirement_occurrence objects.

4.3.501 Ee_requirement_occurrence to Ee_material

Each Ee_requirement_occurrence has required_material defined by exactly one Ee_material. Each Ee_material defines required_material for zero, one, or more Ee_requirement_occurrence objects.

4.3.502 Ee_requirement_occurrence to Characteristic

Each Ee_requirement_occurrence has required_characteristic defined by one or more Characteristic objects. Each Characteristic defines required_characteristic for zero, one, or more Ee_requirement_occurrence objects.

4.3.503 Ee_requirement_occurrence to Ee_specification

Each Ee_requirement_occurrence has requirement_specification defined by one or more Ee_specification objects. Each Ee_specification defines requirement_specification for zero, one, or more Ee_requirement_occurrence objects. Each Ee_requirement_occurrence has required_specification defined by one or more Ee_specification objects. Each Ee_specification defines required_specification for zero, one, or more Ee_requirement_occurrence objects.

4.3.504 Ee_requirement_occurrence to Coordinated_characteristic

Each Ee_requirement_occurrence has required_coordinated_characteristic defined by one or more Coordinated_characteristic objects. Each Coordinated_characteristic defines required_coordinated_characteristic for zero, one, or more Ee_requirement_occurrence objects.

4.3.505 Ee_requirement_occurrence to Ee_product

Each Ee_requirement_occurrence has required_part defined by one or more Ee_product objects. Each Ee_product defines required_part for zero, one, or more Ee_requirement_occurrence objects.

4.3.506 Ee_requirement_occurrence to Functional_specification

Each Ee_requirement_occurrence has required_functional_specification defined by one or more Functional_specification objects. Each Functional_specification defines required_functional_specification for zero, one, or more Ee_requirement_occurrence objects.

4.3.507 Ee_requirement_occurrence to Analytical_representation

Each Ee_requirement_occurrence has required_analytical_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines required_analytical_representation for zero, one, or more Ee_requirement_occurrence objects.

4.3.508 Ee_requirement_occurrence to Requirement_definition

Each Ee_requirement_occurrence has associated_definition defined by exactly one Requirement_definition. Each Requirement_definition defines associated_definition for zero, one, or more Ee_requirement_occurrence objects.

4.3.509 Ee_rule_complex_clause to Ee_rule_simple_clause

Each Ee_rule_complex_clause has conjunctive_content defined by one or more Ee_rule_simple_clause objects. Each Ee_rule_simple_clause defines conjunctive_content for zero, one, or more Ee_rule_complex_clause objects.

4.3.510 Ee_rule_conclusion_definition to Ee_rule_simple_clause

Each Ee_rule_conclusion_definition has conjunctive_content defined by one or more Ee_rule_simple_clause objects. Each Ee_rule_simple_clause defines conjunctive_content for zero, one, or more Ee_rule_conclusion_definition objects.

4.3.511 Ee_rule_conclusion_definition to Ee_rule_definition

Each Ee_rule_conclusion_definition has rule_definition defined by exactly one Ee_rule_definition. Each Ee_rule_definition defines rule_definition for zero, one, or more Ee_rule_conclusion_definition objects.

4.3.512 Ee_rule_definition to Ee_rule_version

Each Ee_rule_definition has defined_versioned_rule defined by exactly one Ee_rule_version. Each Ee_rule_version defines defined_versioned_rule for zero, one, or more Ee_rule_definition objects.

4.3.513 Ee_rule_function to Ee_specification

Each Ee_rule_function has source_code defined by exactly one Ee_specification. Each Ee_specification defines source_code for zero, one, or more Ee_rule_function objects.

4.3.514 Ee_rule_function to Ee_document

Each Ee_rule_function has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Ee_rule_function objects.

4.3.515 Ee_rule_function to Model_parameter

Each Ee_rule_function has parameter defined by one or more Model_parameter objects. Each Model_parameter defines parameter for zero, one, or more Ee_rule_function objects.

4.3.516 Ee_rule_premise_definition to Ee_rule_complex_clause

Each Ee_rule_premise_definition has disjunctive_content defined by one or more Ee_rule_complex_clause objects. Each Ee_rule_complex_clause defines disjunctive_content for zero, one, or more Ee_rule_premise_definition objects.

4.3.517 Ee_rule_premise_definition to Ee_rule_definition

Each Ee_rule_premise_definition has rule_definition defined by exactly one Ee_rule_definition. Each Ee_rule_definition defines rule_definition for zero, one, or more Ee_rule_premise_definition objects.

4.3.518 Ee_rule_simple_clause to Parameter_assignment

Each Ee_rule_simple_clause has parameter defined by one or more Parameter_assignment objects. Each Parameter_assignment defines parameter for zero, one, or more Ee_rule_simple_clause objects.

4.3.519 Ee_rule_simple_clause to Ee_rule_function

Each Ee_rule_simple_clause has rule_function defined by exactly one Ee_rule_function. Each Ee_rule_function defines rule_function for zero, one, or more Ee_rule_simple_clause objects.

4.3.520 Ee_rule_version to Ee_document

Each Ee_rule_version has source defined by zero or one Ee_document objects. Each Ee_document defines source for zero, one, or more Ee_rule_version objects. This assertion is established through a SELECT type.

4.3.521 Ee_rule_version to Organization

Each Ee_rule_version has source defined by zero or one Organization objects. Each Organization defines source for zero, one, or more Ee_rule_version objects. This assertion is established through a SELECT type.

4.3.522 Ee_rule_version to Ee_rule

Each Ee_rule_version has configuration_managed_rule defined by exactly one Ee_rule. Each Ee_rule defines configuration_managed_rule for one or more Ee_rule_version objects.

4.3.523 Ee_specification to Person_and_organization

Each Ee_specification has source defined by zero or one Person_and_organization objects. Each Person_and_organization defines source for zero, one, or more Ee_specification objects. This assertion is established through a SELECT type.

4.3.524 Ee_specification to Organization

Each Ee_specification has source defined by zero or one Organization objects. Each Organization defines source for zero, one, or more Ee_specification objects. This assertion is established through a SELECT type.

4.3.525 Electrical_isolation_laminate_component to Electrical_isolation_removal_template

Each Electrical_isolation_laminate_component has usage_definition defined by exactly one Electrical_isolation_removal_template. Each Electrical_isolation_removal_template defines usage_definition for zero, one, or more Electrical_isolation_laminate_component objects.

4.3.526 Electrical_isolation_removal_structure to Ee_requirement_occurrence

Each Electrical_isolation_removal_structure has electrical_isolation_spacing_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines electrical_isolation_spacing_requirement for zero, one, or more Electrical_isolation_removal_structure objects.

4.3.527 Electrical_isolation_removal_template to Layout_spacing_requirement_occurrence

Each Electrical_isolation_removal_template has electrical_isolation_spacing_requirement defined by exactly one Layout_spacing_requirement_occurrence. Each Layout_spacing_requirement_occurrence defines electrical_isolation_spacing_requirement for zero, one, or more Electrical_isolation_removal_template objects.

4.3.528 Elliptic_area to Length_data_element

Each Elliptic_area has semi_axis_2 defined by exactly one Length_data_element. Each Length_data_element defines semi_axis_2 for zero, one, or more Elliptic_area objects. Each Elliptic_area has semi_axis_1 defined by exactly one Length_data_element. Each Length_data_element defines semi_axis_1 for zero, one, or more Elliptic_area objects.

4.3.529 Elliptic_area to Axis_placement_2d

Each Elliptic_area has position defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines position for zero, one, or more Elliptic_area objects.

4.3.530 Embedded_component_terminal to Packaged_component

Each Embedded_component_terminal has associated_component defined by zero or one Packaged_component objects. Each Packaged_component defines associated_component for zero, one, or more Embedded_component_terminal objects. This assertion is established through a SELECT type.

4.3.531 Embedded_component_terminal to Bare_die_component

Each Embedded_component_terminal has associated_component defined by zero or one Bare_die_component objects. Each Bare_die_component defines associated_component for zero, one, or more Embedded_component_terminal objects. This assertion is established through a SELECT type.

4.3.532 Embedded_component_terminal to Bare_die_terminal

Each Embedded_component_terminal has definition defined by zero or one Bare_die_terminal objects. Each Bare_die_terminal defines definition for zero, one, or more Embedded_component_terminal objects. This assertion is established through a SELECT type.

4.3.533 Embedded_component_terminal to Packaged_part_terminal

Each Embedded_component_terminal has definition defined by zero or one Packaged_part_terminal objects. Each Packaged_part_terminal defines definition for zero, one, or more Embedded_component_terminal objects. This assertion is established through a SELECT type.

4.3.534 Engineering_make_from to Ee_product_definition

Each Engineering_make_from has resultant_design defined by exactly one Ee_product_definition. Each Ee_product_definition defines resultant_design for zero, one, or more Engineering_make_from objects. Each Engineering_make_from has base_design defined by exactly one Ee_product_definition. Each Ee_product_definition defines base_design for zero, one, or more Engineering_make_from objects.

4.3.535 Equivalent_functional_terminals_assignment to Functional_unit_usage_view_terminal_definition

Each Equivalent_functional_terminals_assignment has equivalent_terminals defined by two or more Functional_unit_usage_view_terminal_definition objects. Each Functional_unit_usage_view_terminal_definition defines equivalent_terminals for zero, one, or more Equivalent_functional_terminals_assignment objects.

4.3.536 Equivalent_functional_unit_definition_assignment to Functional_unit_definition

Each Equivalent_functional_unit_definition_assignment has equivalent_functional_unit_definitions defined by two or more Functional_unit_definition objects. Each Functional_unit_definition defines equivalent_functional_unit_definitions for zero, one, or more Equivalent_functional_unit_definition_assignment objects.

4.3.537 Evaluated_characteristic to Coordinated_characteristic

Each Evaluated_characteristic has product_characterization_data defined by one or more Coordinated_characteristic objects. Each Coordinated_characteristic defines product_characterization_data for zero, one, or more Evaluated_characteristic objects.

4.3.538 Evaluated_characteristic to Planned_characteristic

Each Evaluated_characteristic has evaluated_product_data defined by exactly one Planned_characteristic. Each Planned_characteristic defines evaluated_product_data for zero, one, or more Evaluated_characteristic objects.

4.3.539 Externally_defined_dimension_representation to Data_element

Each Externally_defined_dimension_representation has dimension_value defined by exactly one Data_element. Each Data_element defines dimension_value for zero, one, or more Externally_defined_dimension_representation objects.

4.3.540 Externally_defined_dimension_representation to External_definition

Each Externally_defined_dimension_representation has definitions defined by one or more External_definition objects. Each External_definition defines definitions for zero, one, or more Externally_defined_dimension_representation objects.

4.3.541 External_definition to Ee_document

Each External_definition has reference_document defined by zero or one Ee_document objects. Each Ee_document defines reference_document for zero, one, or more External_definition objects. This assertion is established through a SELECT type.

4.3.542 Fabrication_joint to Interconnect_module

Each Fabrication_joint has associated_interconnect_module defined by exactly one Interconnect_module. Each Interconnect_module defines associated_interconnect_module for zero, one, or more Fabrication_joint objects.

4.3.543 Fabrication_joint to Stratum_feature_template_component

Each Fabrication_joint has auxiliary_joint_material defined by exactly one Stratum_feature_template_component. Each Stratum_feature_template_component defines auxiliary_joint_material for zero, one, or more Fabrication_joint objects.

4.3.544 Fabrication_joint to Component_termination_passage_join_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Component_termination_passage_join_terminal objects. Each Component_termination_passage_join_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Component_termination_passage_join_terminal objects. Each Component_termination_passage_join_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.545 Fabrication_joint to Via_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Via_terminal objects. Each Via_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Via_terminal objects. Each Via_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.546 Fabrication_joint to Printed_component_join_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Printed_component_join_terminal objects. Each Printed_component_join_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Printed_component_join_terminal objects. Each Printed_component_join_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.547 Fabrication_joint to Conductive_interconnect_element_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Conductive_interconnect_element_terminal objects. Each Conductive_interconnect_element_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Conductive_interconnect_element_terminal objects. Each Conductive_interconnect_element_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.548 Fabrication_joint to Embedded_component_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Embedded_component_terminal objects. Each Embedded_component_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Embedded_component_terminal objects. Each Embedded_component_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.549 Fabrication_joint to Land_join_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Land_join_terminal objects. Each Land_join_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Land_join_terminal objects. Each Land_join_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.550 Fabrication_joint to Non_functional_land_join_terminal

Each Fabrication_joint has fabrication_features_2 defined by zero or one Non_functional_land_join_terminal objects. Each Non_functional_land_join_terminal defines fabrication_features_2 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type. Each Fabrication_joint has fabrication_features_1 defined by zero or one Non_functional_land_join_terminal objects. Each Non_functional_land_join_terminal defines fabrication_features_1 for zero, one, or more Fabrication_joint objects. This assertion is established through a SELECT type.

4.3.551 Feature_group_relationship to Part_feature

Each Feature_group_relationship has related_feature defined by exactly one Part_feature. Each Part_feature defines related_feature for zero, one, or more Feature_group_relationship objects.

4.3.552 Feature_group_relationship to Part_group_feature

Each Feature_group_relationship has relating_feature defined by exactly one Part_group_feature. Each Part_group_feature defines relating_feature for zero, one, or more Feature_group_relationship objects.

4.3.553 Filled_area_material_removal_laminate_component to Padstack_definition

Each Filled_area_material_removal_laminate_component has reference_padstack defined by exactly one Padstack_definition. Each Padstack_definition defines reference_padstack for zero, one, or more Filled_area_material_removal_laminate_component objects.

4.3.554 Filled_area_material_removal_structure to Conductive_filled_area

Each Filled_area_material_removal_structure has design_intent defined by exactly one Conductive_filled_area. Each Conductive_filled_area defines design_intent for zero, one, or more Filled_area_material_removal_structure objects.

4.3.555 Fill_area to Closed_curve

Each Fill_area has boundaries defined by one or more Closed_curve objects. Each Closed_curve defines boundaries for zero, one, or more Fill_area objects.

4.3.556 Fill_area_occurrence to Curve_style

Each Fill_area_occurrence has boundary_style defined by exactly one Curve_style. Each Curve_style defines boundary_style for zero, one, or more Fill_area_occurrence objects.

4.3.557 Fill_area_occurrence to Cartesian_point

Each Fill_area_occurrence has fill_style_target defined by exactly one Cartesian_point. Each Cartesian_point defines fill_style_target for zero, one, or more Fill_area_occurrence objects.

4.3.558 Fill_area_occurrence to Fill_area

Each Fill_area_occurrence has filled_area defined by exactly one Fill_area. Each Fill_area defines filled_area for zero, one, or more Fill_area_occurrence objects.

4.3.559 Fill_area_occurrence to Fill_area_style

Each Fill_area_occurrence has style defined by one or more Fill_area_style objects. Each Fill_area_style defines style for zero, one, or more Fill_area_occurrence objects.

4.3.560 Fill_area_style to Solid_fill_style

Each Fill_area_style has fill_style defined by zero or one Solid_fill_style objects. Each Solid_fill_style defines fill_style for zero, one, or more Fill_area_style objects. This assertion is established through a SELECT type.

4.3.561 Fill_area_style to Fill_area_style_tile

Each Fill_area_style has fill_style defined by zero or one Fill_area_style_tile objects. Each Fill_area_style_tile defines fill_style for zero, one, or more Fill_area_style objects. This assertion is established through a SELECT type.

4.3.562 Fill_area_style to Externally_defined_tile_style

Each Fill_area_style has fill_style defined by zero or one Externally_defined_tile_style objects. Each Externally_defined_tile_style defines fill_style for zero, one, or more Fill_area_style objects. This assertion is established through a SELECT type.

4.3.563 Fill_area_style to Externally_defined_hatch_style

Each Fill_area_style has fill_style defined by zero or one Externally_defined_hatch_style objects. Each Externally_defined_hatch_style defines fill_style for zero, one, or more Fill_area_style objects. This assertion is established through a SELECT type.

4.3.564 Fill_area_style to Fill_area_style_hatching

Each Fill_area_style has fill_style defined by zero or one Fill_area_style_hatching objects. Each Fill_area_style_hatching defines fill_style for zero, one, or more Fill_area_style objects. This assertion is established through a SELECT type.

4.3.565 Fill_area_style_hatching to Angle_data_element

Each Fill_area_style_hatching has hatch_line_angle defined by exactly one Angle_data_element. Each Angle_data_element defines hatch_line_angle for zero, one, or more Fill_area_style_hatching objects.

4.3.566 Fill_area_style_hatching to Cartesian_point

Each Fill_area_style_hatching has pattern_start defined by exactly one Cartesian_point. Each Cartesian_point defines pattern_start for zero, one, or more Fill_area_style_hatching objects. Each Fill_area_style_hatching has point_of_reference_hatch_line defined by exactly one Cartesian_point. Each Cartesian_point defines point_of_reference_hatch_line for zero, one, or more Fill_area_style_hatching objects.

4.3.567 Fill_area_style_hatching to Vector

Each Fill_area_style_hatching has start_of_next_hatch_line defined by exactly one Vector. Each Vector defines start_of_next_hatch_line for zero, one, or more Fill_area_style_hatching objects.

4.3.568 Fill_area_style_hatching to Curve_style

Each Fill_area_style_hatching has hatch_line_appearance defined by exactly one Curve_style. Each Curve_style defines hatch_line_appearance for zero, one, or more Fill_area_style_hatching objects.

4.3.569 Fill_area_style_tile to Externally_defined_tile

Each Fill_area_style_tile has tile defined by zero or one Externally_defined_tile objects. Each Externally_defined_tile defines tile for zero, one, or more Fill_area_style_tile objects. This assertion is established through a SELECT type.

4.3.570 Fill_area_style_tile to Fill_area_style_tile_symbol_with_style

Each Fill_area_style_tile has tile defined by zero or one Fill_area_style_tile_symbol_with_style objects. Each Fill_area_style_tile_symbol_with_style defines tile for zero, one, or more Fill_area_style_tile objects. This assertion is established through a SELECT type.

4.3.571 Fill_area_style_tile to Fill_area_style_tile_solid_region

Each Fill_area_style_tile has tile defined by zero or one Fill_area_style_tile_solid_region objects. Each Fill_area_style_tile_solid_region defines tile for zero, one, or more Fill_area_style_tile objects. This assertion is established through a SELECT type.

4.3.572 Fill_area_style_tile to Fill_area_style_tile_curve_with_style

Each Fill_area_style_tile has tile defined by zero or one Fill_area_style_tile_curve_with_style objects. Each Fill_area_style_tile_curve_with_style defines tile for zero, one, or more Fill_area_style_tile objects. This assertion is established through a SELECT type.

4.3.573 Fill_area_style_tile to Vector

Each Fill_area_style_tile has second_direction_repeat_factor defined by exactly one Vector. Each Vector defines second_direction_repeat_factor for zero, one, or more Fill_area_style_tile objects. Each Fill_area_style_tile has first_direction_repeat_factor defined by exactly one Vector. Each Vector defines first_direction_repeat_factor for zero, one, or more Fill_area_style_tile objects.

4.3.574 Footprint_definition to Connection_zone_map_identification

Each Footprint_definition has connection_zone_map defined by one or more Connection_zone_map_identification objects. Each Connection_zone_map_identification defines connection_zone_map for zero, one, or more Footprint_definition objects.

4.3.575 Footprint_definition to Reference_packaged_part_interconnect_implementation

Each Footprint_definition has reference_layout defined by exactly one Reference_packaged_part_interconnect_implementation. Each Reference_packaged_part_interconnect_implementation defines reference_layout for zero, one, or more Footprint_definition objects.

4.3.576 Footprint_definition to Assembly_group_component

Each Footprint_definition has reference_breakout defined by exactly one Assembly_group_component. Each Assembly_group_component defines reference_breakout for zero, one, or more Footprint_definition objects.

4.3.577 Footprint_definition to Padstack_definition

Each Footprint_definition has reference_padstack defined by one or more Padstack_definition objects. Each Padstack_definition defines reference_padstack for zero, one, or more Footprint_definition objects.

4.3.578 Footprint_definition to Package

Each Footprint_definition has reference_package defined by exactly one Package. Each Package defines reference_package for zero, one, or more Footprint_definition objects.

4.3.579 Frozen_assignment to Ee_approval

Each Frozen_assignment has assigned_approval defined by exactly one Ee_approval. Each Ee_approval defines assigned_approval for zero, one, or more Frozen_assignment objects.

4.3.580 Frozen_assignment to Frozen_design_object

Each Frozen_assignment has frozen_item defined by one or more Frozen_design_object objects. Each Frozen_design_object defines frozen_item for zero, one, or more Frozen_assignment objects.

4.3.581 Functional_specification to External_definition

Each Functional_specification has characteristic_category defined by exactly one External_definition. Each External_definition defines characteristic_category for zero, one, or more Functional_specification objects.

4.3.582 Functional_specification to Signal_specification

Each Functional_specification has characterizing_signal defined by exactly one Signal_specification. Each Signal_specification defines characterizing_signal for zero, one, or more Functional_specification objects. Each Functional_specification has reference_signal defined by exactly one Signal_specification. Each Signal_specification defines reference_signal for zero, one, or more Functional_specification objects.

4.3.583 Functional_specification to Functional_unit

Each Functional_specification has characterizing_functional_unit defined by exactly one Functional_unit. Each Functional_unit defines characterizing_functional_unit for zero, one, or more Functional_specification objects.

4.3.584 Functional_specification to Functional_specification_definition

Each Functional_specification has associated_definition defined by exactly one Functional_specification_definition. Each Functional_specification_definition defines associated_definition for one or more Functional_specification objects.

4.3.585 Functional_terminal_group_assignment to Scalar_terminal_definition

Each Functional_terminal_group_assignment has functional_usage_view_terminal defined by exactly one Scalar_terminal_definition. Each Scalar_terminal_definition defines functional_usage_view_terminal for zero, one, or more Functional_terminal_group_assignment objects.

4.3.586 Functional_terminal_group_assignment to Functional_terminal_group

Each Functional_terminal_group_assignment has composed_group defined by exactly one Functional_terminal_group. Each Functional_terminal_group defines composed_group for zero, one, or more Functional_terminal_group_assignment objects.

4.3.587 Functional_unit to Parameter_assignment

Each Functional_unit has functional_property defined by one or more Parameter_assignment objects. Each Parameter_assignment defines functional_property for zero, one, or more Functional_unit objects.

4.3.588 Functional_unit to Functional_unit_network_definition

Each Functional_unit has composed_network defined by exactly one Functional_unit_network_definition. Each Functional_unit_network_definition defines composed_network for zero, one, or more Functional_unit objects.

4.3.589 Functional_unit to Functional_unit_definition

Each Functional_unit has definition defined by exactly one Functional_unit_definition. Each Functional_unit_definition defines definition for zero, one, or more Functional_unit objects.

4.3.590 Functional_unit_definition to Analytical_representation

Each Functional_unit_definition has analytical_functional_unit_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines analytical_functional_unit_representation for zero, one, or more Functional_unit_definition objects.

4.3.591 Functional_unit_network_definition to Functional_unit_usage_view

Each Functional_unit_network_definition has usage_view defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines usage_view for zero, one, or more Functional_unit_network_definition objects.

4.3.592 Functional_unit_network_node_definition to Functional_unit_network_definition

Each Functional_unit_network_node_definition has associated_functional_unit_definition defined by exactly one Functional_unit_network_definition. Each Functional_unit_network_definition defines associated_functional_unit_definition for zero, one, or more Functional_unit_network_node_definition objects.

4.3.593 Functional_unit_network_terminal_definition_bus_assignment to Functional_unit_usage_view_terminal_definition

Each Functional_unit_network_terminal_definition_bus_assignment has connected_terminal defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines connected_terminal for zero, one, or more Functional_unit_network_terminal_definition_bus_assignment objects.

4.3.594 Functional_unit_network_terminal_definition_bus_assignment to Bus_structural_definition

Each Functional_unit_network_terminal_definition_bus_assignment has connected_bus defined by exactly one Bus_structural_definition. Each Bus_structural_definition defines connected_bus for zero or one Functional_unit_network_terminal_definition_bus_assignment objects.

4.3.595 Functional_unit_network_terminal_definition_node_assignment to Functional_unit_usage_view_terminal_definition

Each Functional_unit_network_terminal_definition_node_assignment has connected_terminal defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines connected_terminal for zero, one, or more Functional_unit_network_terminal_definition_node_assignment objects.

4.3.596 Functional_unit_network_terminal_definition_node_assignment to Functional_unit_network_node_definition

Each Functional_unit_network_terminal_definition_node_assignment has composed_node defined by exactly one Functional_unit_network_node_definition. Each Functional_unit_network_node_definition defines composed_node for zero or one Functional_unit_network_terminal_definition_node_assignment objects.

4.3.597 Functional_unit_requirement_allocation to Functional_unit

Each Functional_unit_requirement_allocation has functional_unit_assignment defined by exactly one Functional_unit. Each Functional_unit defines functional_unit_assignment for zero, one, or more Functional_unit_requirement_allocation objects.

4.3.598 Functional_unit_requirement_allocation to Ee_requirement_occurrence

Each Functional_unit_requirement_allocation has functional_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines functional_requirement for zero, one, or more Functional_unit_requirement_allocation objects.

4.3.599 Functional_unit_terminal to Functional_unit_usage_view_terminal_definition

Each Functional_unit_terminal has definition defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines definition for zero, one, or more Functional_unit_terminal objects.

4.3.600 Functional_unit_terminal to Functional_unit

Each Functional_unit_terminal has accessed_functional_unit defined by exactly one Functional_unit. Each Functional_unit defines accessed_functional_unit for one or more Functional_unit_terminal objects.

4.3.601 Functional_unit_terminal_bus_assignment to Functional_unit_terminal

Each Functional_unit_terminal_bus_assignment has connected_terminal defined by exactly one Functional_unit_terminal. Each Functional_unit_terminal defines connected_terminal for zero or one Functional_unit_terminal_bus_assignment objects.

4.3.602 Functional_unit_terminal_bus_assignment to Bus_structural_definition

Each Functional_unit_terminal_bus_assignment has connected_bus defined by exactly one Bus_structural_definition. Each Bus_structural_definition defines connected_bus for zero, one, or more Functional_unit_terminal_bus_assignment objects.

4.3.603 Functional_unit_terminal_node_assignment to Functional_unit_terminal

Each Functional_unit_terminal_node_assignment has connected_terminal defined by exactly one Functional_unit_terminal. Each Functional_unit_terminal defines connected_terminal for zero or one Functional_unit_terminal_node_assignment objects.

4.3.604 Functional_unit_terminal_node_assignment to Functional_unit_network_node_definition

Each Functional_unit_terminal_node_assignment has composed_node defined by exactly one Functional_unit_network_node_definition. Each Functional_unit_network_node_definition defines composed_node for zero, one, or more Functional_unit_terminal_node_assignment objects.

4.3.605 Functional_unit_usage_view_terminal_definition to Functional_unit_usage_view

Each Functional_unit_usage_view_terminal_definition has associated_functional_unit_definition defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines associated_functional_unit_definition for one or more Functional_unit_usage_view_terminal_definition objects.

4.3.606 Functional_unit_usage_view_terminal_definition_make_from_relationship to Functional_unit_usage_view_terminal_definition

Each Functional_unit_usage_view_terminal_definition_make_from_relationship has resultant_feature defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines resultant_feature for zero, one, or more Functional_unit_usage_view_terminal_definition_make_from_relationship objects. Each Functional_unit_usage_view_terminal_definition_make_from_relationship has reusable_feature defined by exactly one Functional_unit_usage_view_terminal_definition. Each Functional_unit_usage_view_terminal_definition defines reusable_feature for zero, one, or more Functional_unit_usage_view_terminal_definition_make_from_relationship objects.

4.3.607 Functional_unit_usage_view_terminal_definition_make_from_relationship to Engineering_make_from

Each Functional_unit_usage_view_terminal_definition_make_from_relationship has associated_make_from defined by exactly one Engineering_make_from. Each Engineering_make_from defines associated_make_from for zero, one, or more Functional_unit_usage_view_terminal_definition_make_from_relationship objects.

4.3.608 Functional_usage_view_to_part_terminal_assignment to Scalar_terminal_definition

Each Functional_usage_view_to_part_terminal_assignment has functional_usage_view_terminal defined by exactly one Scalar_terminal_definition. Each Scalar_terminal_definition defines functional_usage_view_terminal for zero, one, or more Functional_usage_view_to_part_terminal_assignment objects.

4.3.609 Functional_usage_view_to_part_terminal_assignment to Part_terminal

Each Functional_usage_view_to_part_terminal_assignment has physical_usage_view_terminal defined by exactly one Part_terminal. Each Part_terminal defines physical_usage_view_terminal for zero or one Functional_usage_view_to_part_terminal_assignment objects.

4.3.610 Functional_usage_view_to_printed_part_template_terminal_assignment to Scalar_terminal_definition

Each Functional_usage_view_to_printed_part_template_terminal_assignment has functional_usage_view_terminal defined by exactly one Scalar_terminal_definition. Each Scalar_terminal_definition defines functional_usage_view_terminal for zero, one, or more Functional_usage_view_to_printed_part_template_terminal_assignment objects.

4.3.611 Functional_usage_view_to_printed_part_template_terminal_assignment to Printed_part_template_terminal

Each Functional_usage_view_to_printed_part_template_terminal_assignment has physical_usage_view_terminal defined by exactly one Printed_part_template_terminal. Each Printed_part_template_terminal defines physical_usage_view_terminal for zero or one Functional_usage_view_to_printed_part_template_terminal_assignment objects.

4.3.612 Gate_path_swap_specification to Design_functional_unit_allocation_to_reference_functional_unit

Each Gate_path_swap_specification has previous_design_object defined by exactly one Design_functional_unit_allocation_to_reference_functional_unit. Each Design_functional_unit_allocation_to_reference_functional_unit defines previous_design_object for zero, one, or more Gate_path_swap_specification objects. Each Gate_path_swap_specification has current_design_object defined by exactly one Design_functional_unit_allocation_to_reference_functional_unit. Each Design_functional_unit_allocation_to_reference_functional_unit defines current_design_object for zero, one, or more Gate_path_swap_specification objects.

4.3.613 Geometric_representation_relationship to Shape_definition

Each Geometric_representation_relationship has shape_2 defined by exactly one Shape_definition. Each Shape_definition defines shape_2 for zero, one, or more Geometric_representation_relationship objects. Each Geometric_representation_relationship has shape_1 defined by exactly one Shape_definition. Each Shape_definition defines shape_1 for zero, one, or more Geometric_representation_relationship objects.

4.3.614 Geometric_tolerance to Centre_axis

Each Geometric_tolerance has toleranced_shape_element defined by zero or one Centre_axis objects. Each Centre_axis defines toleranced_shape_element for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.615 Geometric_tolerance to Centre_plane

Each Geometric_tolerance has toleranced_shape_element defined by zero or one Centre_plane objects. Each Centre_plane defines toleranced_shape_element for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.616 Geometric_tolerance to Centre_point

Each Geometric_tolerance has toleranced_shape_element defined by zero or one Centre_point objects. Each Centre_point defines toleranced_shape_element for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.617 Geometric_tolerance to Tangent_plane

Each Geometric_tolerance has toleranced_shape_element defined by zero or one Tangent_plane objects. Each Tangent_plane defines toleranced_shape_element for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.618 Geometric_tolerance to Physical_feature_or_part_template

Each Geometric_tolerance has toleranced_shape_element defined by zero or one Physical_feature_or_part_template objects. Each Physical_feature_or_part_template defines toleranced_shape_element for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.619 Geometric_tolerance to Tolerance_zone_definition

Each Geometric_tolerance has tolerance_zone_or_boundary_definition defined by zero or one Tolerance_zone_definition objects. Each Tolerance_zone_definition defines tolerance_zone_or_boundary_definition for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.620 Geometric_tolerance to Profile_related_positional_boundary_definition

Each Geometric_tolerance has tolerance_zone_or_boundary_definition defined by zero or one Profile_related_positional_boundary_definition objects. Each Profile_related_positional_boundary_definition defines tolerance_zone_or_boundary_definition for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.621 Geometric_tolerance to Positional_boundary_member_definition

Each Geometric_tolerance has tolerance_zone_or_boundary_definition defined by zero or one Positional_boundary_member_definition objects. Each Positional_boundary_member_definition defines tolerance_zone_or_boundary_definition for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.622 Geometric_tolerance to Conical_tolerance_zone_boundary_diametrical_size_characteristic

Each Geometric_tolerance has tolerance_zone_or_boundary_definition defined by zero or one Conical_tolerance_zone_boundary_diametrical_size_characteristic objects. Each Conical_tolerance_zone_boundary_diametrical_size_characteristic defines tolerance_zone_or_boundary_definition for zero, one, or more Geometric_tolerance objects. This assertion is established through a SELECT type.

4.3.623 Geometric_tolerance_qualifier to Geometric_tolerance

Each Geometric_tolerance_qualifier has qualified_geometric_tolerances defined by one or more Geometric_tolerance objects. Each Geometric_tolerance defines qualified_geometric_tolerances for zero, one, or more Geometric_tolerance_qualifier objects.

4.3.624 Geometric_tolerance_with_referenced_datum_system to Datum_system_defining_conditions

Each Geometric_tolerance_with_referenced_datum_system has datum_system_definition defined by exactly one Datum_system_defining_conditions. Each Datum_system_defining_conditions defines datum_system_definition for zero, one, or more Geometric_tolerance_with_referenced_datum_system objects.

4.3.625 Interconnect_component_interface_terminal to Connection_zone

Each Interconnect_component_interface_terminal has connection defined by exactly one Connection_zone. Each Connection_zone defines connection for zero, one, or more Interconnect_component_interface_terminal objects.

4.3.626 Interconnect_component_join_terminal to Physical_component

Each Interconnect_component_join_terminal has associated_component defined by exactly one Physical_component. Each Physical_component defines associated_component for zero, one, or more Interconnect_component_join_terminal objects.

4.3.627 Interconnect_component_join_terminal to Connection_zone

Each Interconnect_component_join_terminal has connection defined by exactly one Connection_zone. Each Connection_zone defines connection for zero, one, or more Interconnect_component_join_terminal objects.

4.3.628 Interconnect_module to Assembly_module

Each Interconnect_module has assembly_design_requirement defined by one or more Assembly_module objects. Each Assembly_module defines assembly_design_requirement for zero, one, or more Interconnect_module objects.

4.3.629 Interconnect_module_cavity_surface to Interconnect_module_surface_feature

Each Interconnect_module_cavity_surface has reference_surface defined by exactly one Interconnect_module_surface_feature. Each Interconnect_module_surface_feature defines reference_surface for zero, one, or more Interconnect_module_cavity_surface objects.

4.3.630 Interconnect_module_component to Interconnect_module_usage_view

Each Interconnect_module_component has usage_definition defined by exactly one Interconnect_module_usage_view. Each Interconnect_module_usage_view defines usage_definition for zero, one, or more Interconnect_module_component objects.

4.3.631 Interconnect_module_component_stratum_based_terminal to Interconnect_module_stratum_based_terminal

Each Interconnect_module_component_stratum_based_terminal has definition defined by exactly one Interconnect_module_stratum_based_terminal. Each Interconnect_module_stratum_based_terminal defines definition for zero, one, or more Interconnect_module_component_stratum_based_terminal objects.

4.3.632 Interconnect_module_component_surface_feature to Interconnect_module_surface_feature

Each Interconnect_module_component_surface_feature has definition defined by exactly one Interconnect_module_surface_feature. Each Interconnect_module_surface_feature defines definition for zero, one, or more Interconnect_module_component_surface_feature objects.

4.3.633 Interconnect_module_component_terminal to Interconnect_module_component

Each Interconnect_module_component_terminal has associated_component defined by exactly one Interconnect_module_component. Each Interconnect_module_component defines associated_component for two or more Interconnect_module_component_terminal objects.

4.3.634 Interconnect_module_component_terminal to Interconnect_module_terminal

Each Interconnect_module_component_terminal has definition defined by exactly one Interconnect_module_terminal. Each Interconnect_module_terminal defines definition for zero, one, or more Interconnect_module_component_terminal objects.

4.3.635 Interconnect_module_constraint_region to Inter_stratum_extent

Each Interconnect_module_constraint_region has associated_stratum_extent defined by one or more Inter_stratum_extent objects. Each Inter_stratum_extent defines associated_stratum_extent for zero, one, or more Interconnect_module_constraint_region objects.

4.3.636 Interconnect_module_constraint_region to Ee_requirement_occurrence

Each Interconnect_module_constraint_region has requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines requirement for zero, one, or more Interconnect_module_constraint_region objects.

4.3.637 Interconnect_module_constraint_region to Interconnect_module

Each Interconnect_module_constraint_region has scope defined by exactly one Interconnect_module. Each Interconnect_module defines scope for zero, one, or more Interconnect_module_constraint_region objects.

4.3.638 Interconnect_module_constraint_region to Via

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Via objects. Each Via defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.639 Interconnect_module_constraint_region to Cutout

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Cutout objects. Each Cutout defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.640 Interconnect_module_constraint_region to Component_termination_passage

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Component_termination_passage objects. Each Component_termination_passage defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.641 Interconnect_module_constraint_region to Land

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Land objects. Each Land defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.642 Interconnect_module_constraint_region to Conductor

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Conductor objects. Each Conductor defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.643 Interconnect_module_constraint_region to Conductive_filled_area

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Conductive_filled_area objects. Each Conductive_filled_area defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.644 Interconnect_module_constraint_region to Stratum_feature

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.645 Interconnect_module_constraint_region to Laminate_component

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Laminate_component objects. Each Laminate_component defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.646 Interconnect_module_constraint_region to Inter_stratum_feature

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.647 Interconnect_module_constraint_region to Non_functional_land

Each Interconnect_module_constraint_region has non_conformant_interconnect_module_design_object defined by zero or one Non_functional_land objects. Each Non_functional_land defines non_conformant_interconnect_module_design_object for zero, one, or more Interconnect_module_constraint_region objects. This assertion is established through a SELECT type.

4.3.648 Interconnect_module_cutout_segment_surface to Edge_segment_vertex

Each Interconnect_module_cutout_segment_surface has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Interconnect_module_cutout_segment_surface objects. Each Interconnect_module_cutout_segment_surface has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Interconnect_module_cutout_segment_surface objects.

4.3.649 Interconnect_module_cutout_segment_surface to Interconnect_module_cutout_surface

Each Interconnect_module_cutout_segment_surface has composed_surface defined by exactly one Interconnect_module_cutout_surface. Each Interconnect_module_cutout_surface defines composed_surface for zero, one, or more Interconnect_module_cutout_segment_surface objects.

4.3.650 Interconnect_module_edge_segment to Inter_stratum_feature_edge_segment_template

Each Interconnect_module_edge_segment has usage_definition defined by exactly one Inter_stratum_feature_edge_segment_template. Each Inter_stratum_feature_edge_segment_template defines usage_definition for zero, one, or more Interconnect_module_edge_segment objects.

4.3.651 Interconnect_module_edge_segment to Edge_segment_vertex

Each Interconnect_module_edge_segment has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Interconnect_module_edge_segment objects. Each Interconnect_module_edge_segment has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Interconnect_module_edge_segment objects.

4.3.652 Interconnect_module_edge_segment to Interconnect_module_edge

Each Interconnect_module_edge_segment has composed_edge defined by exactly one Interconnect_module_edge. Each Interconnect_module_edge defines composed_edge for zero, one, or more Interconnect_module_edge_segment objects.

4.3.653 Interconnect_module_edge_segment_surface to Edge_segment_vertex

Each Interconnect_module_edge_segment_surface has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Interconnect_module_edge_segment_surface objects. Each Interconnect_module_edge_segment_surface has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Interconnect_module_edge_segment_surface objects.

4.3.654 Interconnect_module_edge_segment_surface to Interconnect_module_edge_surface

Each Interconnect_module_edge_segment_surface has composed_surface defined by exactly one Interconnect_module_edge_surface. Each Interconnect_module_edge_surface defines composed_surface for zero, one, or more Interconnect_module_edge_segment_surface objects.

4.3.655 Interconnect_module_macro_component to Interconnect_module

Each Interconnect_module_macro_component has design_definition defined by exactly one Interconnect_module. Each Interconnect_module defines design_definition for zero, one, or more Interconnect_module_macro_component objects.

4.3.656 Interconnect_module_macro_component_join_terminal to Interconnect_module_macro_component

Each Interconnect_module_macro_component_join_terminal has associated_component defined by exactly one Interconnect_module_macro_component. Each Interconnect_module_macro_component defines associated_component for one or more Interconnect_module_macro_component_join_terminal objects.

4.3.657 Interconnect_module_macro_component_join_terminal to Interconnect_module_join_terminal

Each Interconnect_module_macro_component_join_terminal has definition defined by exactly one Interconnect_module_join_terminal. Each Interconnect_module_join_terminal defines definition for zero, one, or more Interconnect_module_macro_component_join_terminal objects.

4.3.658 Interconnect_module_product_assembly_view_fabrication_joint to Interconnect_module_component_stratum_based_terminal

Each Interconnect_module_product_assembly_view_fabrication_joint has joined_terminal defined by one or more Interconnect_module_component_stratum_based_terminal objects. Each Interconnect_module_component_stratum_based_terminal defines joined_terminal for zero, one, or more Interconnect_module_product_assembly_view_fabrication_joint objects.

4.3.659 Interconnect_module_stratum_assembly_relationship to Stratum

Each Interconnect_module_stratum_assembly_relationship has component defined by exactly one Stratum. Each Stratum defines component for one or more Interconnect_module_stratum_assembly_relationship objects.

4.3.660 Interconnect_module_stratum_assembly_relationship to Interconnect_module

Each Interconnect_module_stratum_assembly_relationship has assembly defined by exactly one Interconnect_module. Each Interconnect_module defines assembly for zero, one, or more Interconnect_module_stratum_assembly_relationship objects.

4.3.661 Interconnect_module_surface_feature to Interconnect_module_usage_view

Each Interconnect_module_surface_feature has associated_definition defined by exactly one Interconnect_module_usage_view. Each Interconnect_module_usage_view defines associated_definition for zero, one, or more Interconnect_module_surface_feature objects.

4.3.662 Interconnect_module_terminal to Interconnect_module_usage_view

Each Interconnect_module_terminal has associated_definition defined by exactly one Interconnect_module_usage_view. Each Interconnect_module_usage_view defines associated_definition for zero, one, or more Interconnect_module_terminal objects.

4.3.663 Interconnect_module_terminal to Connection_zone

Each Interconnect_module_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Interconnect_module_terminal objects.

4.3.664 Interconnect_module_terminal_surface_constituent_relationship to Interconnect_module_terminal

Each Interconnect_module_terminal_surface_constituent_relationship has related_shape_element defined by exactly one Interconnect_module_terminal. Each Interconnect_module_terminal defines related_shape_element for zero, one, or more Interconnect_module_terminal_surface_constituent_relationship objects.

4.3.665 Interconnect_module_terminal_surface_constituent_relationship to Interconnect_module_surface_feature

Each Interconnect_module_terminal_surface_constituent_relationship has relating_shape_element defined by exactly one Interconnect_module_surface_feature. Each Interconnect_module_surface_feature defines relating_shape_element for zero, one, or more Interconnect_module_terminal_surface_constituent_relationship objects.

4.3.666 Interconnect_module_usage_view to Functional_unit_usage_view

Each Interconnect_module_usage_view has implemented_function defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines implemented_function for zero, one, or more Interconnect_module_usage_view objects.

4.3.667 Interconnect_module_usage_view to Located_interconnect_module_thickness_requirement

Each Interconnect_module_usage_view has located_thickness_requirement defined by one or more Located_interconnect_module_thickness_requirement objects. Each Located_interconnect_module_thickness_requirement defines located_thickness_requirement for one or more Interconnect_module_usage_view objects.

4.3.668 Interconnect_module_usage_view to Restraint_condition

Each Interconnect_module_usage_view has measurement_condition defined by exactly one Restraint_condition. Each Restraint_condition defines measurement_condition for zero, one, or more Interconnect_module_usage_view objects.

4.3.669 Interconnect_module_usage_view to Length_data_element

Each Interconnect_module_usage_view has maximum_thickness_over_dielectric_requirement defined by exactly one Length_data_element. Each Length_data_element defines maximum_thickness_over_dielectric_requirement for zero, one, or more Interconnect_module_usage_view objects. Each Interconnect_module_usage_view has minimum_thickness_over_dielectric_requirement defined by exactly one Length_data_element. Each Length_data_element defines minimum_thickness_over_dielectric_requirement for zero, one, or more Interconnect_module_usage_view objects. Each Interconnect_module_usage_view has maximum_thickness_over_metal_requirement defined by exactly one Length_data_element. Each Length_data_element defines maximum_thickness_over_metal_requirement for zero, one, or more Interconnect_module_usage_view objects. Each Interconnect_module_usage_view has minimum_thickness_over_metal_requirement defined by exactly one Length_data_element. Each Length_data_element defines minimum_thickness_over_metal_requirement for zero, one, or more Interconnect_module_usage_view objects.

4.3.670 Interfaced_group_component to Interface_component

Each Interfaced_group_component has group_interface_component defined by one or more Interface_component objects. Each Interface_component defines group_interface_component for zero, one, or more Interfaced_group_component objects.

4.3.671 Interface_access_material_removal_laminate_component to Laminate_component_interface_terminal

Each Interface_access_material_removal_laminate_component has accessed_interface defined by exactly one Laminate_component_interface_terminal. Each Laminate_component_interface_terminal defines accessed_interface for zero, one, or more Interface_access_material_removal_laminate_component objects.

4.3.672 Interface_access_stratum_feature_template_component to Laminate_component_interface_terminal

Each Interface_access_stratum_feature_template_component has accessed_interface defined by exactly one Laminate_component_interface_terminal. Each Laminate_component_interface_terminal defines accessed_interface for zero, one, or more Interface_access_stratum_feature_template_component objects.

4.3.673 Interface_component_allocation to Mating_connector_component

Each Interface_component_allocation has mating_requirement defined by exactly one Mating_connector_component. Each Mating_connector_component defines mating_requirement for zero, one, or more Interface_component_allocation objects.

4.3.674 Interface_component_allocation to Interface_component

Each Interface_component_allocation has mating_solution defined by exactly one Interface_component. Each Interface_component defines mating_solution for zero, one, or more Interface_component_allocation objects.

4.3.675 Interface_mounted_join to Physical_component_terminal

Each Interface_mounted_join has access_to_interface_mounted_component defined by exactly one Physical_component_terminal. Each Physical_component_terminal defines access_to_interface_mounted_component for zero, one, or more Interface_mounted_join objects.

4.3.676 Interface_mounted_join to Packaged_connector_component_interface_terminal

Each Interface_mounted_join has connection_to_interconnect_module_component defined by exactly one Packaged_connector_component_interface_terminal. Each Packaged_connector_component_interface_terminal defines connection_to_interconnect_module_component for zero, one, or more Interface_mounted_join objects.

4.3.677 Interface_requirement_occurrence to Next_higher_assembly_interface_requirement

Each Interface_requirement_occurrence has interface_to_next_higher_assembly defined by exactly one Next_higher_assembly_interface_requirement. Each Next_higher_assembly_interface_requirement defines interface_to_next_higher_assembly for zero, one, or more Interface_requirement_occurrence objects.

4.3.678 Internal_probe_access_area to Stratum_feature

Each Internal_probe_access_area has stratum_feature_implementation defined by exactly one Stratum_feature. Each Stratum_feature defines stratum_feature_implementation for zero, one, or more Internal_probe_access_area objects.

4.3.679 Internal_stratum_access to Via

Each Internal_stratum_access has inter_stratum_feature defined by exactly one Via. Each Via defines inter_stratum_feature for zero, one, or more Internal_stratum_access objects.

4.3.680 Internal_stratum_access to Internal_probe_access_area

Each Internal_stratum_access has connected_probe defined by exactly one Internal_probe_access_area. Each Internal_probe_access_area defines connected_probe for zero, one, or more Internal_stratum_access objects.

4.3.681 Inter_stratum_extent to Stratum

Each Inter_stratum_extent has to_stratum defined by exactly one Stratum. Each Stratum defines to_stratum for zero, one, or more Inter_stratum_extent objects. Each Inter_stratum_extent has from_stratum defined by exactly one Stratum. Each Stratum defines from_stratum for zero, one, or more Inter_stratum_extent objects.

4.3.682 Inter_stratum_feature to Inter_stratum_feature_template

Each Inter_stratum_feature has usage_definition defined by exactly one Inter_stratum_feature_template. Each Inter_stratum_feature_template defines usage_definition for zero, one, or more Inter_stratum_feature objects.

4.3.683 Inter_stratum_feature_edge_segment_template to Edge_segment_cross_section

Each Inter_stratum_feature_edge_segment_template has cross_section defined by exactly one Edge_segment_cross_section. Each Edge_segment_cross_section defines cross_section for zero, one, or more Inter_stratum_feature_edge_segment_template objects.

4.3.684 Inter_stratum_feature_edge_segment_template to Trimmed_line

Each Inter_stratum_feature_edge_segment_template has reference_path defined by exactly one Trimmed_line. Each Trimmed_line defines reference_path for zero, one, or more Inter_stratum_feature_edge_segment_template objects.

4.3.685 Inter_stratum_feature_edge_segment_template to Edge_segment_vertex

Each Inter_stratum_feature_edge_segment_template has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Inter_stratum_feature_edge_segment_template objects. Each Inter_stratum_feature_edge_segment_template has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Inter_stratum_feature_edge_segment_template objects.

4.3.686 Inter_stratum_feature_edge_segment_template to Inter_stratum_feature_edge_template

Each Inter_stratum_feature_edge_segment_template has composed_edge defined by exactly one Inter_stratum_feature_edge_template. Each Inter_stratum_feature_edge_template defines composed_edge for two or more Inter_stratum_feature_edge_segment_template objects.

4.3.687 Inter_stratum_feature_template to Passage_technology

Each Inter_stratum_feature_template has of_passage_technology defined by exactly one Passage_technology. Each Passage_technology defines of_passage_technology for zero, one, or more Inter_stratum_feature_template objects.

4.3.688 Inter_stratum_join_implementation to Inter_stratum_join_relationship

Each Inter_stratum_join_implementation has topological_requirement defined by exactly one Inter_stratum_join_relationship. Each Inter_stratum_join_relationship defines topological_requirement for zero, one, or more Inter_stratum_join_implementation objects.

4.3.689 Inter_stratum_join_relationship to Layer_connection_point

Each Inter_stratum_join_relationship has points_to_be_connected defined by two or more Layer_connection_point objects. Each Layer_connection_point defines points_to_be_connected for zero or one Inter_stratum_join_relationship objects.

4.3.690 Intra_stratum_join_relationship to Layer_connection_point

Each Intra_stratum_join_relationship has points_to_be_connected defined by two or more Layer_connection_point objects. Each Layer_connection_point defines points_to_be_connected for zero, one, or more Intra_stratum_join_relationship objects.

4.3.691 Item_restricted_requirement_occurrence to Laminate_component

Each Item_restricted_requirement_occurrence has basis defined by zero or one Laminate_component objects. Each Laminate_component defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.692 Item_restricted_requirement_occurrence to Inter_stratum_feature

Each Item_restricted_requirement_occurrence has basis defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.693 Item_restricted_requirement_occurrence to Stratum_feature

Each Item_restricted_requirement_occurrence has basis defined by zero or one Stratum_feature objects. Each Stratum_feature defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.694 Item_restricted_requirement_occurrence to Stratum

Each Item_restricted_requirement_occurrence has basis defined by zero or one Stratum objects. Each Stratum defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.695 Item_restricted_requirement_occurrence to Stratum_surface

Each Item_restricted_requirement_occurrence has basis defined by zero or one Stratum_surface objects. Each Stratum_surface defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.696 Item_restricted_requirement_occurrence to Ee_requirement_occurrence

Each Item_restricted_requirement_occurrence has basis defined by zero or one Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.697 Item_restricted_requirement_occurrence to Part_feature

Each Item_restricted_requirement_occurrence has basis defined by zero or one Part_feature objects. Each Part_feature defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.698 Item_restricted_requirement_occurrence to Assembly_component

Each Item_restricted_requirement_occurrence has basis defined by zero or one Assembly_component objects. Each Assembly_component defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.699 Item_restricted_requirement_occurrence to Component_external_reference

Each Item_restricted_requirement_occurrence has basis defined by zero or one Component_external_reference objects. Each Component_external_reference defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.700 Item_restricted_requirement_occurrence to Component_feature

Each Item_restricted_requirement_occurrence has basis defined by zero or one Component_feature objects. Each Component_feature defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.701 Item_restricted_requirement_occurrence to Component_feature_external_reference

Each Item_restricted_requirement_occurrence has basis defined by zero or one Component_feature_external_reference objects. Each Component_feature_external_reference defines basis for zero, one, or more Item_restricted_requirement_occurrence objects. This assertion is established through a SELECT type.

4.3.702 Join_relationship to Interconnect_module

Each Join_relationship has associated_layout defined by exactly one Interconnect_module. Each Interconnect_module defines associated_layout for zero, one, or more Join_relationship objects.

4.3.703 Join_two_physical_connectivity_definition_supporting_inter_stratum_feature to Physical_connectivity_definition

Each Join_two_physical_connectivity_definition_supporting_inter_stratum_feature has joined_definitions defined by exactly two Physical_connectivity_definition objects. Each Physical_connectivity_definition defines joined_definitions for zero, one, or more Join_two_physical_connectivity_definition_supporting_inter_stratum_feature objects.

4.3.704 Join_two_physical_connectivity_definition_supporting_printed_component to Physical_connectivity_definition

Each Join_two_physical_connectivity_definition_supporting_printed_component has joined_definitions defined by exactly two Physical_connectivity_definition objects. Each Physical_connectivity_definition defines joined_definitions for zero, one, or more Join_two_physical_connectivity_definition_supporting_printed_component objects.

4.3.705 Junction_vertex_allocation to Topological_junction

Each Junction_vertex_allocation has required_vertex defined by exactly one Topological_junction. Each Topological_junction defines required_vertex for zero, one, or more Junction_vertex_allocation objects.

4.3.706 Junction_vertex_allocation to Inter_stratum_join_relationship

Each Junction_vertex_allocation has implementation_point defined by zero or one Inter_stratum_join_relationship objects. Each Inter_stratum_join_relationship defines implementation_point for zero, one, or more Junction_vertex_allocation objects. This assertion is established through a SELECT type.

4.3.707 Junction_vertex_allocation to Explicitly_located_layer_connection_point

Each Junction_vertex_allocation has implementation_point defined by zero or one Explicitly_located_layer_connection_point objects. Each Explicitly_located_layer_connection_point defines implementation_point for zero, one, or more Junction_vertex_allocation objects. This assertion is established through a SELECT type.

4.3.708 Laminate_component to Part_template

Each Laminate_component has usage_definition defined by exactly one Part_template. Each Part_template defines usage_definition for zero, one, or more Laminate_component objects.

4.3.709 Laminate_component to External_definition

Each Laminate_component has cad_layer defined by exactly one External_definition. Each External_definition defines cad_layer for zero, one, or more Laminate_component objects.

4.3.710 Laminate_group_component_make_from_relationship to Stratum_make_from_relationship

Each Laminate_group_component_make_from_relationship has stratum_mapping defined by one or more Stratum_make_from_relationship objects. Each Stratum_make_from_relationship defines stratum_mapping for zero, one, or more Laminate_group_component_make_from_relationship objects.

4.3.711 Laminate_group_component_make_from_relationship to Assembly_group_component

Each Laminate_group_component_make_from_relationship has resultant_component defined by exactly one Assembly_group_component. Each Assembly_group_component defines resultant_component for zero, one, or more Laminate_group_component_make_from_relationship objects. Each Laminate_group_component_make_from_relationship has reusable_component defined by exactly one Assembly_group_component. Each Assembly_group_component defines reusable_component for zero, one, or more Laminate_group_component_make_from_relationship objects.

4.3.712 Laminate_text_component to Stratum_feature

Each Laminate_text_component has stratum_feature_implementation defined by one or more Stratum_feature objects. Each Stratum_feature defines stratum_feature_implementation for zero, one, or more Laminate_text_component objects.

4.3.713 Laminate_text_component to Part_text_template

Each Laminate_text_component has usage_definition defined by exactly one Part_text_template. Each Part_text_template defines usage_definition for zero, one, or more Laminate_text_component objects.

4.3.714 Land to Land_physical_template

Each Land has usage_definition defined by exactly one Land_physical_template. Each Land_physical_template defines usage_definition for zero, one, or more Land objects. Each Land has alternate_land_definition defined by exactly one Land_physical_template. Each Land_physical_template defines alternate_land_definition for zero, one, or more Land objects.

4.3.715 Land_interface_terminal to Land

Each Land_interface_terminal has associated_component defined by exactly one Land. Each Land defines associated_component for zero, one, or more Land_interface_terminal objects.

4.3.716 Land_interface_terminal to Land_template_interface_terminal

Each Land_interface_terminal has definition defined by exactly one Land_template_interface_terminal. Each Land_template_interface_terminal defines definition for zero, one, or more Land_interface_terminal objects.

4.3.717 Land_join_terminal to Land

Each Land_join_terminal has associated_component defined by exactly one Land. Each Land defines associated_component for one or more Land_join_terminal objects.

4.3.718 Land_join_terminal to Land_template_join_terminal

Each Land_join_terminal has definition defined by exactly one Land_template_join_terminal. Each Land_template_join_terminal defines definition for zero, one, or more Land_join_terminal objects.

4.3.719 Land_physical_template to External_definition

Each Land_physical_template has pre_defined_classification_code defined by exactly one External_definition. Each External_definition defines pre_defined_classification_code for zero, one, or more Land_physical_template objects.

4.3.720 Land_template_terminal to Connection_zone

Each Land_template_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Land_template_terminal objects.

4.3.721 Land_template_terminal to Land_physical_template

Each Land_template_terminal has associated_definition defined by exactly one Land_physical_template. Each Land_physical_template defines associated_definition for two or more Land_template_terminal objects.

4.3.722 Layer to External_definition

Each Layer has layer_usage defined by exactly one External_definition. Each External_definition defines layer_usage for zero, one, or more Layer objects.

4.3.723 Layer to Inter_stratum_extent

Each Layer has extent defined by exactly one Inter_stratum_extent. Each Inter_stratum_extent defines extent for zero, one, or more Layer objects.

4.3.724 Layer_connection_point to Design_layer_stratum

Each Layer_connection_point has resident_design_layer_stratum defined by exactly one Design_layer_stratum. Each Design_layer_stratum defines resident_design_layer_stratum for one or more Layer_connection_point objects.

4.3.725 Layer_connection_point to Cartesian_point

Each Layer_connection_point has location_3d defined by exactly one Cartesian_point. Each Cartesian_point defines location_3d for zero, one, or more Layer_connection_point objects. Each Layer_connection_point has location_2d defined by exactly one Cartesian_point. Each Cartesian_point defines location_2d for zero, one, or more Layer_connection_point objects.

4.3.726 Layer_connection_point_link to Layer_connection_point

Each Layer_connection_point_link has subsequent_point defined by exactly one Layer_connection_point. Each Layer_connection_point defines subsequent_point for zero, one, or more Layer_connection_point_link objects. Each Layer_connection_point_link has precedent_point defined by exactly one Layer_connection_point. Each Layer_connection_point defines precedent_point for zero, one, or more Layer_connection_point_link objects.

4.3.727 Layer_connection_point_link to Stratum

Each Layer_connection_point_link has associated_stratum defined by exactly one Stratum. Each Stratum defines associated_stratum for zero, one, or more Layer_connection_point_link objects.

4.3.728 Layer_qualified_layout_spacing_requirement_occurrence to Inter_stratum_extent

Each Layer_qualified_layout_spacing_requirement_occurrence has layer_context defined by one or more Inter_stratum_extent objects. Each Inter_stratum_extent defines layer_context for zero, one, or more Layer_qualified_layout_spacing_requirement_occurrence objects.

4.3.729 Layout_land_width_tolerance_requirement_occurrence to Length_data_element

Each Layout_land_width_tolerance_requirement_occurrence has maximum_negative_deviation defined by exactly one Length_data_element. Each Length_data_element defines maximum_negative_deviation for zero, one, or more Layout_land_width_tolerance_requirement_occurrence objects.

4.3.730 Layout_land_width_tolerance_requirement_occurrence to Pitch_class

Each Layout_land_width_tolerance_requirement_occurrence has mating_feature_pitch_class defined by exactly one Pitch_class. Each Pitch_class defines mating_feature_pitch_class for zero, one, or more Layout_land_width_tolerance_requirement_occurrence objects.

4.3.731 Layout_line_width_tolerance_requirement_occurrence to Length_data_element

Each Layout_line_width_tolerance_requirement_occurrence has maximum_positive_deviation defined by exactly one Length_data_element. Each Length_data_element defines maximum_positive_deviation for zero, one, or more Layout_line_width_tolerance_requirement_occurrence objects. Each Layout_line_width_tolerance_requirement_occurrence has maximum_negative_deviation defined by exactly one Length_data_element. Each Length_data_element defines maximum_negative_deviation for zero, one, or more Layout_line_width_tolerance_requirement_occurrence objects.

4.3.732 Layout_line_width_tolerance_requirement_occurrence to Line_width_class

Each Layout_line_width_tolerance_requirement_occurrence has tolerance_line_width_class defined by exactly one Line_width_class. Each Line_width_class defines tolerance_line_width_class for zero, one, or more Layout_line_width_tolerance_requirement_occurrence objects.

4.3.733 Layout_line_width_tolerance_requirement_occurrence to Stratum_technology

Each Layout_line_width_tolerance_requirement_occurrence has characterized_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines characterized_stratum_technology for zero, one, or more Layout_line_width_tolerance_requirement_occurrence objects.

4.3.734 Layout_macro_component to Layout_macro_floor_plan_template

Each Layout_macro_component has usage_definition defined by exactly one Layout_macro_floor_plan_template. Each Layout_macro_floor_plan_template defines usage_definition for zero, one, or more Layout_macro_component objects.

4.3.735 Layout_macro_component to Layout_macro_definition

Each Layout_macro_component has design_definition defined by exactly one Layout_macro_definition. Each Layout_macro_definition defines design_definition for zero, one, or more Layout_macro_component objects.

4.3.736 Layout_macro_definition to Layout_macro_floor_plan_template

Each Layout_macro_definition has floor_layout_view defined by exactly one Layout_macro_floor_plan_template. Each Layout_macro_floor_plan_template defines floor_layout_view for zero or one Layout_macro_definition objects.

4.3.737 Layout_macro_definition_terminal_to_usage_terminal_assignment to Component_termination_passage_join_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Component_termination_passage_join_terminal objects. Each Component_termination_passage_join_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.738 Layout_macro_definition_terminal_to_usage_terminal_assignment to Via_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Via_terminal objects. Each Via_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.739 Layout_macro_definition_terminal_to_usage_terminal_assignment to Printed_component_join_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Printed_component_join_terminal objects. Each Printed_component_join_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.740 Layout_macro_definition_terminal_to_usage_terminal_assignment to Conductive_interconnect_element_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Conductive_interconnect_element_terminal objects. Each Conductive_interconnect_element_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.741 Layout_macro_definition_terminal_to_usage_terminal_assignment to Embedded_component_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Embedded_component_terminal objects. Each Embedded_component_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.742 Layout_macro_definition_terminal_to_usage_terminal_assignment to Land_join_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Land_join_terminal objects. Each Land_join_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.743 Layout_macro_definition_terminal_to_usage_terminal_assignment to Non_functional_land_join_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has assigned_design_object defined by zero or one Non_functional_land_join_terminal objects. Each Non_functional_land_join_terminal defines assigned_design_object for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. This assertion is established through a SELECT type.

4.3.744 Layout_macro_definition_terminal_to_usage_terminal_assignment to Printed_part_template_terminal

Each Layout_macro_definition_terminal_to_usage_terminal_assignment has defined_template_feature defined by exactly one Printed_part_template_terminal. Each Printed_part_template_terminal defines defined_template_feature for zero, one, or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects.

4.3.745 Layout_macro_floor_plan_template to Layout_macro_definition_terminal_to_usage_terminal_assignment

Each Layout_macro_floor_plan_template has access_maps defined by one or more Layout_macro_definition_terminal_to_usage_terminal_assignment objects. Each Layout_macro_definition_terminal_to_usage_terminal_assignment defines access_maps for zero, one, or more Layout_macro_floor_plan_template objects.

4.3.746 Layout_spacing_requirement_non_conforming_design_object to Layout_spacing_requirement_occurrence

Each Layout_spacing_requirement_non_conforming_design_object has design_requirement defined by exactly one Layout_spacing_requirement_occurrence. Each Layout_spacing_requirement_occurrence defines design_requirement for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects.

4.3.747 Layout_spacing_requirement_non_conforming_design_object to Assembly_component

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Assembly_component objects. Each Assembly_component defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Assembly_component objects. Each Assembly_component defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.748 Layout_spacing_requirement_non_conforming_design_object to Component_feature

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Component_feature objects. Each Component_feature defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Component_feature objects. Each Component_feature defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.749 Layout_spacing_requirement_non_conforming_design_object to Via

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Via objects. Each Via defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Via objects. Each Via defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.750 Layout_spacing_requirement_non_conforming_design_object to Cutout

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Cutout objects. Each Cutout defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Cutout objects. Each Cutout defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.751 Layout_spacing_requirement_non_conforming_design_object to Component_termination_passage

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Component_termination_passage objects. Each Component_termination_passage defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Component_termination_passage objects. Each Component_termination_passage defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.752 Layout_spacing_requirement_non_conforming_design_object to Land

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Land objects. Each Land defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Land objects. Each Land defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.753 Layout_spacing_requirement_non_conforming_design_object to Conductor

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Conductor objects. Each Conductor defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Conductor objects. Each Conductor defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.754 Layout_spacing_requirement_non_conforming_design_object to Conductive_filled_area

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Conductive_filled_area objects. Each Conductive_filled_area defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Conductive_filled_area objects. Each Conductive_filled_area defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.755 Layout_spacing_requirement_non_conforming_design_object to Stratum_feature

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.756 Layout_spacing_requirement_non_conforming_design_object to Laminate_component

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Laminate_component objects. Each Laminate_component defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Laminate_component objects. Each Laminate_component defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.757 Layout_spacing_requirement_non_conforming_design_object to Inter_stratum_feature

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.758 Layout_spacing_requirement_non_conforming_design_object to Non_functional_land

Each Layout_spacing_requirement_non_conforming_design_object has reference_design_object defined by zero or one Non_functional_land objects. Each Non_functional_land defines reference_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type. Each Layout_spacing_requirement_non_conforming_design_object has non_conformant_design_object defined by zero or one Non_functional_land objects. Each Non_functional_land defines non_conformant_design_object for zero, one, or more Layout_spacing_requirement_non_conforming_design_object objects. This assertion is established through a SELECT type.

4.3.759 Layout_spacing_requirement_occurrence to Minimum_tolerance_characteristic

Each Layout_spacing_requirement_occurrence has minimum_spacing defined by exactly one Minimum_tolerance_characteristic. Each Minimum_tolerance_characteristic defines minimum_spacing for zero, one, or more Layout_spacing_requirement_occurrence objects.

4.3.760 Length_trimmed_terminal to Length_data_element

Each Length_trimmed_terminal has minimum_trimmed_length defined by exactly one Length_data_element. Each Length_data_element defines minimum_trimmed_length for zero, one, or more Length_trimmed_terminal objects. Each Length_trimmed_terminal has maximum_trimmed_length defined by exactly one Length_data_element. Each Length_data_element defines maximum_trimmed_length for zero, one, or more Length_trimmed_terminal objects.

4.3.761 Limit_dimension_representation to Data_element

Each Limit_dimension_representation has lower_dimension_value defined by exactly one Data_element. Each Data_element defines lower_dimension_value for zero, one, or more Limit_dimension_representation objects. Each Limit_dimension_representation has upper_dimension_value defined by exactly one Data_element. Each Data_element defines upper_dimension_value for zero, one, or more Limit_dimension_representation objects.

4.3.762 Linear_array_placement_group_component to Length_data_element

Each Linear_array_placement_group_component has terminus_node_positive_x_displacement defined by exactly one Length_data_element. Each Length_data_element defines terminus_node_positive_x_displacement for zero, one, or more Linear_array_placement_group_component objects. Each Linear_array_placement_group_component has initial_node_positive_x_displacement defined by exactly one Length_data_element. Each Length_data_element defines initial_node_positive_x_displacement for zero, one, or more Linear_array_placement_group_component objects.

4.3.763 Linear_array_placement_group_component to Assembly_component_placement_link

Each Linear_array_placement_group_component has element defined by one or more Assembly_component_placement_link objects. Each Assembly_component_placement_link defines element for zero, one, or more Linear_array_placement_group_component objects.

4.3.764 Linear_array_placement_group_component_link to Linear_array_placement_group_component

Each Linear_array_placement_group_component_link has subsequent_element defined by exactly one Linear_array_placement_group_component. Each Linear_array_placement_group_component defines subsequent_element for zero, one, or more Linear_array_placement_group_component_link objects. Each Linear_array_placement_group_component_link has precedent_element defined by exactly one Linear_array_placement_group_component. Each Linear_array_placement_group_component defines precedent_element for zero, one, or more Linear_array_placement_group_component_link objects.

4.3.765 Line_width_class to Range_characteristic

Each Line_width_class has class_member defined by exactly one Range_characteristic. Each Range_characteristic defines class_member for zero, one, or more Line_width_class objects.

4.3.766 List_characteristic to Characteristic

Each List_characteristic has values defined by one or more Characteristic objects. Each Characteristic defines values for zero, one, or more List_characteristic objects.

4.3.767 Located_interconnect_module_thickness_requirement to Length_data_element

Each Located_interconnect_module_thickness_requirement has minimum_thickness_requirement defined by exactly one Length_data_element. Each Length_data_element defines minimum_thickness_requirement for zero, one, or more Located_interconnect_module_thickness_requirement objects. Each Located_interconnect_module_thickness_requirement has maximum_thickness_requirement defined by exactly one Length_data_element. Each Length_data_element defines maximum_thickness_requirement for zero, one, or more Located_interconnect_module_thickness_requirement objects.

4.3.768 Located_interconnect_module_thickness_requirement to Axis_placement

Each Located_interconnect_module_thickness_requirement has reference_location defined by exactly one Axis_placement. Each Axis_placement defines reference_location for zero, one, or more Located_interconnect_module_thickness_requirement objects.

4.3.769 Marking to Ee_specification

Each Marking has marking_specification defined by exactly one Ee_specification. Each Ee_specification defines marking_specification for zero, one, or more Marking objects.

4.3.770 Marking to Styled_text

Each Marking has message_representation_text defined by exactly one Styled_text. Each Styled_text defines message_representation_text for zero, one, or more Marking objects.

4.3.771 Marking to Feature_shape_definition

Each Marking has message_representation defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines message_representation for zero, one, or more Marking objects.

4.3.772 Marking to Axis_placement

Each Marking has placement defined by exactly one Axis_placement. Each Axis_placement defines placement for zero, one, or more Marking objects.

4.3.773 Marking to Ee_material

Each Marking has marking_material defined by exactly one Ee_material. Each Ee_material defines marking_material for zero, one, or more Marking objects.

4.3.774 Marking to Part_template_3d_shape

Each Marking has marked_device_shape defined by zero or one Part_template_3d_shape objects. Each Part_template_3d_shape defines marked_device_shape for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.775 Marking to Physical_unit_3d_shape

Each Marking has marked_device_shape defined by zero or one Physical_unit_3d_shape objects. Each Physical_unit_3d_shape defines marked_device_shape for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.776 Marking to Part_template_non_planar_2d_shape

Each Marking has marked_device_shape defined by zero or one Part_template_non_planar_2d_shape objects. Each Part_template_non_planar_2d_shape defines marked_device_shape for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.777 Marking to Part_template_planar_shape

Each Marking has marked_device_shape defined by zero or one Part_template_planar_shape objects. Each Part_template_planar_shape defines marked_device_shape for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.778 Marking to Physical_unit_planar_shape

Each Marking has marked_device_shape defined by zero or one Physical_unit_planar_shape objects. Each Physical_unit_planar_shape defines marked_device_shape for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.779 Marking to Configuration_controlled_message

Each Marking has marked_message defined by zero or one Configuration_controlled_message objects. Each Configuration_controlled_message defines marked_message for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.780 Marking to Ee_text

Each Marking has marked_message defined by zero or one Ee_text objects. Each Ee_text defines marked_message for zero, one, or more Marking objects. This assertion is established through a SELECT type.

4.3.781 Material_addition_feature to Material_addition_feature_template

Each Material_addition_feature has material_addition_shape defined by exactly one Material_addition_feature_template. Each Material_addition_feature_template defines material_addition_shape for zero, one, or more Material_addition_feature objects.

4.3.782 Material_addition_feature_template to Stratum_technology

Each Material_addition_feature_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Material_addition_feature_template objects.

4.3.783 Material_assembly_relationship to Ee_material

Each Material_assembly_relationship has assembly defined by exactly one Ee_material. Each Ee_material defines assembly for zero, one, or more Material_assembly_relationship objects. Each Material_assembly_relationship has component defined by exactly one Ee_material. Each Ee_material defines component for zero, one, or more Material_assembly_relationship objects.

4.3.784 Material_composition_relationship to Ee_material

Each Material_composition_relationship has relating_product defined by exactly one Ee_material. Each Ee_material defines relating_product for zero, one, or more Material_composition_relationship objects. Each Material_composition_relationship has related_product defined by exactly one Ee_material. Each Ee_material defines related_product for zero, one, or more Material_composition_relationship objects.

4.3.785 Material_composition_relationship to Data_element

Each Material_composition_relationship has constituent_amount defined by exactly one Data_element. Each Data_element defines constituent_amount for zero, one, or more Material_composition_relationship objects.

4.3.786 Material_composition_relationship to Ee_text

Each Material_composition_relationship has determination_method defined by exactly one Ee_text. Each Ee_text defines determination_method for zero, one, or more Material_composition_relationship objects.

4.3.787 Material_removal_feature to Material_removal_feature_template

Each Material_removal_feature has material_removal_shape defined by exactly one Material_removal_feature_template. Each Material_removal_feature_template defines material_removal_shape for zero, one, or more Material_removal_feature objects.

4.3.788 Material_removal_feature_template to Stratum_technology

Each Material_removal_feature_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Material_removal_feature_template objects.

4.3.789 Material_removal_laminate_component to Stratum_feature

Each Material_removal_laminate_component has design_intent defined by exactly one Stratum_feature. Each Stratum_feature defines design_intent for zero, one, or more Material_removal_laminate_component objects.

4.3.790 Material_removal_laminate_component to Material_removal_feature_template

Each Material_removal_laminate_component has usage_definition defined by exactly one Material_removal_feature_template. Each Material_removal_feature_template defines usage_definition for zero, one, or more Material_removal_laminate_component objects.

4.3.791 Mating_connector_component to Ee_specification

Each Mating_connector_component has definition defined by zero or one Ee_specification objects. Each Ee_specification defines definition for zero, one, or more Mating_connector_component objects. This assertion is established through a SELECT type.

4.3.792 Mating_connector_component to Printed_connector_template

Each Mating_connector_component has definition defined by zero or one Printed_connector_template objects. Each Printed_connector_template defines definition for zero, one, or more Mating_connector_component objects. This assertion is established through a SELECT type.

4.3.793 Mating_connector_component to Packaged_connector

Each Mating_connector_component has definition defined by zero or one Packaged_connector objects. Each Packaged_connector defines definition for zero, one, or more Mating_connector_component objects. This assertion is established through a SELECT type.

4.3.794 Mating_connector_placement_relationship to Axis_placement

Each Mating_connector_placement_relationship has connector_placement defined by exactly one Axis_placement. Each Axis_placement defines connector_placement for zero, one, or more Mating_connector_placement_relationship objects.

4.3.795 Mating_connector_placement_relationship to Next_higher_assembly_interface_requirement

Each Mating_connector_placement_relationship has interface_context defined by exactly one Next_higher_assembly_interface_requirement. Each Next_higher_assembly_interface_requirement defines interface_context for zero, one, or more Mating_connector_placement_relationship objects.

4.3.796 Mating_connector_placement_relationship to Mating_connector_component

Each Mating_connector_placement_relationship has placed_connector defined by exactly one Mating_connector_component. Each Mating_connector_component defines placed_connector for zero, one, or more Mating_connector_placement_relationship objects.

4.3.797 Mating_connector_termination to Ee_requirement_occurrence

Each Mating_connector_termination has signal_definition defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines signal_definition for zero, one, or more Mating_connector_termination objects.

4.3.798 Mating_connector_termination to Mating_connector_component

Each Mating_connector_termination has accessed_connector defined by exactly one Mating_connector_component. Each Mating_connector_component defines accessed_connector for zero, one, or more Mating_connector_termination objects.

4.3.799 Mating_connector_termination to Packaged_part_terminal

Each Mating_connector_termination has definition defined by zero or one Packaged_part_terminal objects. Each Packaged_part_terminal defines definition for zero, one, or more Mating_connector_termination objects. This assertion is established through a SELECT type.

4.3.800 Mating_connector_termination to Printed_part_template_terminal

Each Mating_connector_termination has definition defined by zero or one Printed_part_template_terminal objects. Each Printed_part_template_terminal defines definition for zero, one, or more Mating_connector_termination objects. This assertion is established through a SELECT type.

4.3.801 Maximum_tolerance_characteristic to Numeric_parameter

Each Maximum_tolerance_characteristic has component_characteristic defined by exactly one Numeric_parameter. Each Numeric_parameter defines component_characteristic for zero, one, or more Maximum_tolerance_characteristic objects.

4.3.802 Minimally_defined_bare_die_terminal to Bare_die

Each Minimally_defined_bare_die_terminal has associated_definition defined by exactly one Bare_die. Each Bare_die defines associated_definition for two or more Minimally_defined_bare_die_terminal objects.

4.3.803 Minimally_defined_component_terminal to Part_terminal_external_reference

Each Minimally_defined_component_terminal has definition defined by exactly one Part_terminal_external_reference. Each Part_terminal_external_reference defines definition for zero, one, or more Minimally_defined_component_terminal objects.

4.3.804 Minimum_tolerance_characteristic to Numeric_parameter

Each Minimum_tolerance_characteristic has component_characteristic defined by exactly one Numeric_parameter. Each Numeric_parameter defines component_characteristic for zero, one, or more Minimum_tolerance_characteristic objects.

4.3.805 Model_parameter to Coordinated_characteristic

Each Model_parameter has default_value defined by zero or one Coordinated_characteristic objects. Each Coordinated_characteristic defines default_value for zero, one, or more Model_parameter objects. This assertion is established through a SELECT type. Each Model_parameter has valid_range defined by one or more Coordinated_characteristic objects. Each Coordinated_characteristic defines valid_range for zero, one, or more Model_parameter objects.

4.3.806 Model_parameter to Characteristic

Each Model_parameter has default_value defined by zero or one Characteristic objects. Each Characteristic defines default_value for zero, one, or more Model_parameter objects. This assertion is established through a SELECT type.

4.3.807 Model_parameter to Ee_document

Each Model_parameter has reference_document defined by exactly one Ee_document. Each Ee_document defines reference_document for zero, one, or more Model_parameter objects.

4.3.808 Mounting_restriction_area to Assembly_module

Each Mounting_restriction_area has scope defined by exactly one Assembly_module. Each Assembly_module defines scope for zero, one, or more Mounting_restriction_area objects.

4.3.809 Mounting_restriction_area to Interconnect_module_component_surface_feature

Each Mounting_restriction_area has mounting_surface defined by exactly one Interconnect_module_component_surface_feature. Each Interconnect_module_component_surface_feature defines mounting_surface for zero, one, or more Mounting_restriction_area objects.

4.3.810 Mounting_restriction_volume to Assembly_module

Each Mounting_restriction_volume has scope defined by exactly one Assembly_module. Each Assembly_module defines scope for zero, one, or more Mounting_restriction_volume objects.

4.3.811 Mounting_restriction_volume to Interconnect_module_component_surface_feature

Each Mounting_restriction_volume has mounting_surface defined by exactly one Interconnect_module_component_surface_feature. Each Interconnect_module_component_surface_feature defines mounting_surface for zero, one, or more Mounting_restriction_volume objects.

4.3.812 Movable_packaged_component_join_terminal to Length_data_element

Each Movable_packaged_component_join_terminal has minimum_wire_terminal_length defined by exactly one Length_data_element. Each Length_data_element defines minimum_wire_terminal_length for zero, one, or more Movable_packaged_component_join_terminal objects. Each Movable_packaged_component_join_terminal has maximum_wire_terminal_length defined by exactly one Length_data_element. Each Length_data_element defines maximum_wire_terminal_length for zero, one, or more Movable_packaged_component_join_terminal objects.

4.3.813 Multi_layer_material_removal_laminate_component to Layer

Each Multi_layer_material_removal_laminate_component has derived_layer_geometry defined by exactly one Layer. Each Layer defines derived_layer_geometry for zero, one, or more Multi_layer_material_removal_laminate_component objects.

4.3.814 Multi_layer_stratum_feature to Layer

Each Multi_layer_stratum_feature has derived_layer_geometry defined by exactly one Layer. Each Layer defines derived_layer_geometry for zero, one, or more Multi_layer_stratum_feature objects.

4.3.815 Multi_layer_stratum_feature_template_component to Layer

Each Multi_layer_stratum_feature_template_component has derived_layer_geometry defined by exactly one Layer. Each Layer defines derived_layer_geometry for zero, one, or more Multi_layer_stratum_feature_template_component objects.

4.3.816 Next_higher_assembly_interface_requirement to Ee_product_version

Each Next_higher_assembly_interface_requirement has version_of_next_higher_assembly defined by exactly one Ee_product_version. Each Ee_product_version defines version_of_next_higher_assembly for zero, one, or more Next_higher_assembly_interface_requirement objects.

4.3.817 Next_higher_assembly_interface_requirement to Bound_volume_shape

Each Next_higher_assembly_interface_requirement has shape defined by exactly one Bound_volume_shape. Each Bound_volume_shape defines shape for zero, one, or more Next_higher_assembly_interface_requirement objects.

4.3.818 Next_higher_assembly_relationship to Data_element

Each Next_higher_assembly_relationship has quantity defined by exactly one Data_element. Each Data_element defines quantity for zero, one, or more Next_higher_assembly_relationship objects.

4.3.819 Non_conductive_cross_section_template to Length_data_element

Each Non_conductive_cross_section_template has nominal_width defined by exactly one Length_data_element. Each Length_data_element defines nominal_width for zero, one, or more Non_conductive_cross_section_template objects.

4.3.820 Non_conductive_cross_section_template to Stratum_technology

Each Non_conductive_cross_section_template has template_technology defined by exactly one Stratum_technology. Each Stratum_technology defines template_technology for zero, one, or more Non_conductive_cross_section_template objects.

4.3.821 Non_feature_shape_definition to Feature_shape_definition

Each Non_feature_shape_definition has model_shape defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines model_shape for zero, one, or more Non_feature_shape_definition objects.

4.3.822 Non_feature_shape_definition to Non_feature_shape_element

Each Non_feature_shape_definition has associated_element defined by exactly one Non_feature_shape_element. Each Non_feature_shape_element defines associated_element for zero, one, or more Non_feature_shape_definition objects.

4.3.823 Non_feature_shape_element to Footprint_definition

Each Non_feature_shape_element has scope defined by zero or one Footprint_definition objects. Each Footprint_definition defines scope for zero, one, or more Non_feature_shape_element objects. This assertion is established through a SELECT type.

4.3.824 Non_feature_shape_element to Padstack_definition

Each Non_feature_shape_element has scope defined by zero or one Padstack_definition objects. Each Padstack_definition defines scope for zero, one, or more Non_feature_shape_element objects. This assertion is established through a SELECT type.

4.3.825 Non_feature_shape_element to Physical_unit

Each Non_feature_shape_element has scope defined by zero or one Physical_unit objects. Each Physical_unit defines scope for zero, one, or more Non_feature_shape_element objects. This assertion is established through a SELECT type.

4.3.826 Non_feature_shape_element to Stratum

Each Non_feature_shape_element has scope defined by zero or one Stratum objects. Each Stratum defines scope for zero, one, or more Non_feature_shape_element objects. This assertion is established through a SELECT type.

4.3.827 Non_feature_shape_element to Template_definition

Each Non_feature_shape_element has scope defined by zero or one Template_definition objects. Each Template_definition defines scope for zero, one, or more Non_feature_shape_element objects. This assertion is established through a SELECT type.

4.3.828 Non_feature_shape_element_relationship to Non_feature_shape_element

Each Non_feature_shape_element_relationship has relating_shape_element defined by exactly one Non_feature_shape_element. Each Non_feature_shape_element defines relating_shape_element for zero, one, or more Non_feature_shape_element_relationship objects. Each Non_feature_shape_element_relationship has related_shape_element defined by exactly one Non_feature_shape_element. Each Non_feature_shape_element defines related_shape_element for zero, one, or more Non_feature_shape_element_relationship objects.

4.3.829 Non_functional_land to Land_physical_template

Each Non_functional_land has usage_definition defined by exactly one Land_physical_template. Each Land_physical_template defines usage_definition for zero, one, or more Non_functional_land objects. Each Non_functional_land has alternate_land_definition defined by exactly one Land_physical_template. Each Land_physical_template defines alternate_land_definition for zero, one, or more Non_functional_land objects.

4.3.830 Non_functional_land_interface_terminal to Non_functional_land

Each Non_functional_land_interface_terminal has associated_component defined by exactly one Non_functional_land. Each Non_functional_land defines associated_component for zero, one, or more Non_functional_land_interface_terminal objects.

4.3.831 Non_functional_land_interface_terminal to Land_template_interface_terminal

Each Non_functional_land_interface_terminal has definition defined by exactly one Land_template_interface_terminal. Each Land_template_interface_terminal defines definition for zero, one, or more Non_functional_land_interface_terminal objects.

4.3.832 Non_functional_land_join_terminal to Non_functional_land

Each Non_functional_land_join_terminal has associated_component defined by exactly one Non_functional_land. Each Non_functional_land defines associated_component for one or more Non_functional_land_join_terminal objects.

4.3.833 Non_functional_land_join_terminal to Land_template_join_terminal

Each Non_functional_land_join_terminal has definition defined by exactly one Land_template_join_terminal. Each Land_template_join_terminal defines definition for zero, one, or more Non_functional_land_join_terminal objects.

4.3.834 Numeric_parameter to Data_element

Each Numeric_parameter has property_value defined by exactly one Data_element. Each Data_element defines property_value for zero, one, or more Numeric_parameter objects.

4.3.835 Operational_requirement_occurrence_relationship to Ee_requirement_occurrence

Each Operational_requirement_occurrence_relationship has related_requirement_occurrence defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines related_requirement_occurrence for zero, one, or more Operational_requirement_occurrence_relationship objects. Each Operational_requirement_occurrence_relationship has relating_requirement_occurrence defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines relating_requirement_occurrence for zero, one, or more Operational_requirement_occurrence_relationship objects.

4.3.836 Opposing_boundary_set_size_characteristic to Tolerance_zone_opposing_boundary_set

Each Opposing_boundary_set_size_characteristic has sized_boundary_set defined by exactly one Tolerance_zone_opposing_boundary_set. Each Tolerance_zone_opposing_boundary_set defines sized_boundary_set for zero, one, or more Opposing_boundary_set_size_characteristic objects.

4.3.837 Owner_assignment to Person_and_organization

Each Owner_assignment has assigned_owner defined by exactly one Person_and_organization. Each Person_and_organization defines assigned_owner for zero, one, or more Owner_assignment objects.

4.3.838 Owner_assignment to Owned_design_object

Each Owner_assignment has owned_item defined by one or more Owned_design_object objects. Each Owned_design_object defines owned_item for zero, one, or more Owner_assignment objects.

4.3.839 Package to Datum_usage_in_datum_system_relationship

Each Package has of_derived_datum_system_definining_relationship defined by one, two, or three Datum_usage_in_datum_system_relationship objects. Each Datum_usage_in_datum_system_relationship defines of_derived_datum_system_definining_relationship for zero, one, or more Package objects.

4.3.840 Package to Datum_feature_usage_in_datum_system_relationship

Each Package has of_defining_datum_system_relationship defined by one, two, or three Datum_feature_usage_in_datum_system_relationship objects. Each Datum_feature_usage_in_datum_system_relationship defines of_defining_datum_system_relationship for zero, one, or more Package objects.

4.3.841 Package to Datum_reference_frame

Each Package has of_datum_reference_frame defined by exactly one Datum_reference_frame. Each Datum_reference_frame defines of_datum_reference_frame for zero, one, or more Package objects.

4.3.842 Package to Length_data_element

Each Package has least_lead_length_below_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines least_lead_length_below_seating_plane for zero, one, or more Package objects. Each Package has maximum_lead_length_below_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_lead_length_below_seating_plane for zero, one, or more Package objects. Each Package has minimum_body_clearance_below_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines minimum_body_clearance_below_seating_plane for zero, one, or more Package objects. Each Package has minimum_body_clearance_above_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines minimum_body_clearance_above_seating_plane for zero, one, or more Package objects. Each Package has maximum_body_clearance_below_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_body_clearance_below_seating_plane for zero, one, or more Package objects. Each Package has maximum_body_clearance_above_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_body_clearance_above_seating_plane for zero, one, or more Package objects. Each Package has maximum_body_height_below_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_body_height_below_seating_plane for zero, one, or more Package objects. Each Package has maximum_body_height_above_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines maximum_body_height_above_seating_plane for zero, one, or more Package objects.

4.3.843 Package to Cartesian_point

Each Package has maximum_material_condition_centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines maximum_material_condition_centroid_location for zero, one, or more Package objects. Each Package has least_material_condition_centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines least_material_condition_centroid_location for zero, one, or more Package objects.

4.3.844 Package to Seating_plane

Each Package has package_seating_plane defined by exactly one Seating_plane. Each Seating_plane defines package_seating_plane for zero, one, or more Package objects.

4.3.845 Package to Registered_style

Each Package has case_style defined by one or more Registered_style objects. Each Registered_style defines case_style for zero, one, or more Package objects.

4.3.846 Packaged_component to Altered_package

Each Packaged_component has selected_package_alternate defined by exactly one Altered_package. Each Altered_package defines selected_package_alternate for zero, one, or more Packaged_component objects.

4.3.847 Packaged_component to Packaged_part

Each Packaged_component has usage_definition defined by exactly one Packaged_part. Each Packaged_part defines usage_definition for zero, one, or more Packaged_component objects.

4.3.848 Packaged_component_join_terminal to Packaged_component

Each Packaged_component_join_terminal has associated_component defined by exactly one Packaged_component. Each Packaged_component defines associated_component for one or more Packaged_component_join_terminal objects.

4.3.849 Packaged_component_join_terminal to Packaged_part_join_terminal

Each Packaged_component_join_terminal has definition defined by exactly one Packaged_part_join_terminal. Each Packaged_part_join_terminal defines definition for zero, one, or more Packaged_component_join_terminal objects.

4.3.850 Packaged_connector to Seating_plane

Each Packaged_connector has interface_plane defined by exactly one Seating_plane. Each Seating_plane defines interface_plane for zero, one, or more Packaged_connector objects.

4.3.851 Packaged_connector_component to Packaged_connector

Each Packaged_connector_component has usage_definition defined by exactly one Packaged_connector. Each Packaged_connector defines usage_definition for zero, one, or more Packaged_connector_component objects.

4.3.852 Packaged_connector_component_interface_terminal to Packaged_connector_component

Each Packaged_connector_component_interface_terminal has associated_component defined by exactly one Packaged_connector_component. Each Packaged_connector_component defines associated_component for one or more Packaged_connector_component_interface_terminal objects.

4.3.853 Packaged_connector_component_interface_terminal to Packaged_part_interface_terminal

Each Packaged_connector_component_interface_terminal has definition defined by exactly one Packaged_part_interface_terminal. Each Packaged_part_interface_terminal defines definition for zero, one, or more Packaged_connector_component_interface_terminal objects.

4.3.854 Packaged_connector_terminal_relationship to Packaged_part_interface_terminal

Each Packaged_connector_terminal_relationship has interface_terminal defined by exactly one Packaged_part_interface_terminal. Each Packaged_part_interface_terminal defines interface_terminal for one or more Packaged_connector_terminal_relationship objects.

4.3.855 Packaged_connector_terminal_relationship to Packaged_part_join_terminal

Each Packaged_connector_terminal_relationship has join_terminal defined by exactly one Packaged_part_join_terminal. Each Packaged_part_join_terminal defines join_terminal for zero or one Packaged_connector_terminal_relationship objects.

4.3.856 Packaged_connector_terminal_relationship to Packaged_connector

Each Packaged_connector_terminal_relationship has connector defined by exactly one Packaged_connector. Each Packaged_connector defines connector for one or more Packaged_connector_terminal_relationship objects.

4.3.857 Packaged_part to Functional_unit_usage_view

Each Packaged_part has implemented_function defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines implemented_function for zero, one, or more Packaged_part objects.

4.3.858 Packaged_part to Ee_material

Each Packaged_part has potting_compound defined by exactly one Ee_material. Each Ee_material defines potting_compound for zero, one, or more Packaged_part objects.

4.3.859 Packaged_part to Package

Each Packaged_part has used_package defined by zero or one Package objects. Each Package defines used_package for zero, one, or more Packaged_part objects. This assertion is established through a SELECT type.

4.3.860 Packaged_part to Package_external_reference

Each Packaged_part has used_package defined by zero or one Package_external_reference objects. Each Package_external_reference defines used_package for zero, one, or more Packaged_part objects. This assertion is established through a SELECT type.

4.3.861 Packaged_part_terminal to Packaged_part

Each Packaged_part_terminal has associated_definition defined by exactly one Packaged_part. Each Packaged_part defines associated_definition for zero, one, or more Packaged_part_terminal objects.

4.3.862 Packaged_part_terminal to Package_terminal

Each Packaged_part_terminal has terminal_of_package defined by zero or one Package_terminal objects. Each Package_terminal defines terminal_of_package for zero, one, or more Packaged_part_terminal objects. This assertion is established through a SELECT type.

4.3.863 Packaged_part_terminal to Package_terminal_external_reference

Each Packaged_part_terminal has terminal_of_package defined by zero or one Package_terminal_external_reference objects. Each Package_terminal_external_reference defines terminal_of_package for zero, one, or more Packaged_part_terminal objects. This assertion is established through a SELECT type.

4.3.864 Package_body to Package

Each Package_body has associated_definition defined by exactly one Package. Each Package defines associated_definition for zero or one Package_body objects.

4.3.865 Package_body to Ee_material

Each Package_body has body_material defined by exactly one Ee_material. Each Ee_material defines body_material for zero, one, or more Package_body objects.

4.3.866 Package_body_edge_segment_surface to Edge_segment_vertex

Each Package_body_edge_segment_surface has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Package_body_edge_segment_surface objects. Each Package_body_edge_segment_surface has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Package_body_edge_segment_surface objects.

4.3.867 Package_body_edge_segment_surface to Package_body_edge_surface

Each Package_body_edge_segment_surface has composed_surface defined by exactly one Package_body_edge_surface. Each Package_body_edge_surface defines composed_surface for zero, one, or more Package_body_edge_segment_surface objects.

4.3.868 Package_body_surface to Package_body

Each Package_body_surface has associated_package_body defined by exactly one Package_body. Each Package_body defines associated_package_body for zero, one, or more Package_body_surface objects.

4.3.869 Package_terminal to Connection_zone_interface_plane_relationship

Each Package_terminal has seating_plane_zone defined by exactly one Connection_zone_interface_plane_relationship. Each Connection_zone_interface_plane_relationship defines seating_plane_zone for zero or one Package_terminal objects.

4.3.870 Package_terminal to Length_data_element

Each Package_terminal has maximum_terminal_diametrical_extent defined by exactly one Length_data_element. Each Length_data_element defines maximum_terminal_diametrical_extent for zero, one, or more Package_terminal objects. Each Package_terminal has minimum_terminal_diametrical_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_terminal_diametrical_extent for zero, one, or more Package_terminal objects.

4.3.871 Package_terminal to Material_assembly_relationship

Each Package_terminal has terminal_core_material defined by exactly one Material_assembly_relationship. Each Material_assembly_relationship defines terminal_core_material for zero, one, or more Package_terminal objects. Each Package_terminal has terminal_surface_material defined by exactly one Material_assembly_relationship. Each Material_assembly_relationship defines terminal_surface_material for zero, one, or more Package_terminal objects.

4.3.872 Package_terminal to Characteristic

Each Package_terminal has terminal_characteristic defined by one or more Characteristic objects. Each Characteristic defines terminal_characteristic for zero, one, or more Package_terminal objects.

4.3.873 Package_terminal to Analytical_representation

Each Package_terminal has model_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines model_representation for zero, one, or more Package_terminal objects.

4.3.874 Package_terminal to Connection_zone

Each Package_terminal has internal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines internal_connection_zone for zero, one, or more Package_terminal objects. Each Package_terminal has external_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines external_connection_zone for zero, one, or more Package_terminal objects.

4.3.875 Package_terminal to Package

Each Package_terminal has associated_definition defined by exactly one Package. Each Package defines associated_definition for zero, one, or more Package_terminal objects.

4.3.876 Package_terminal_external_reference to Package_external_reference

Each Package_terminal_external_reference has associated_definition defined by exactly one Package_external_reference. Each Package_external_reference defines associated_definition for zero, one, or more Package_terminal_external_reference objects.

4.3.877 Package_terminal_occurrence to Packaged_component

Each Package_terminal_occurrence has associated_component defined by exactly one Packaged_component. Each Packaged_component defines associated_component for zero, one, or more Package_terminal_occurrence objects.

4.3.878 Package_terminal_occurrence to Package_terminal

Each Package_terminal_occurrence has definition defined by exactly one Package_terminal. Each Package_terminal defines definition for zero, one, or more Package_terminal_occurrence objects.

4.3.879 Package_terminal_surface_constituent_relationship to Package_terminal

Each Package_terminal_surface_constituent_relationship has related_shape_element defined by exactly one Package_terminal. Each Package_terminal defines related_shape_element for zero, one, or more Package_terminal_surface_constituent_relationship objects.

4.3.880 Package_terminal_surface_constituent_relationship to Package_body_surface

Each Package_terminal_surface_constituent_relationship has relating_shape_element defined by exactly one Package_body_surface. Each Package_body_surface defines relating_shape_element for zero, one, or more Package_terminal_surface_constituent_relationship objects.

4.3.881 Padstack_definition to Package_terminal

Each Padstack_definition has reference_package_terminal defined by one or more Package_terminal objects. Each Package_terminal defines reference_package_terminal for zero, one, or more Padstack_definition objects.

4.3.882 Padstack_definition to Template_stratum_assignment

Each Padstack_definition has specific_stratum_assignment defined by one or more Template_stratum_assignment objects. Each Template_stratum_assignment defines specific_stratum_assignment for zero, one, or more Padstack_definition objects.

4.3.883 Padstack_definition to Stratum_feature_template

Each Padstack_definition has of_stratum_feature_template defined by one or more Stratum_feature_template objects. Each Stratum_feature_template defines of_stratum_feature_template for zero, one, or more Padstack_definition objects.

4.3.884 Padstack_definition to Dependent_material_removal_feature_template

Each Padstack_definition has isolation_template defined by one or more Dependent_material_removal_feature_template objects. Each Dependent_material_removal_feature_template defines isolation_template for zero, one, or more Padstack_definition objects.

4.3.885 Padstack_definition to Default_land_physical_template

Each Padstack_definition has land_template defined by one or more Default_land_physical_template objects. Each Default_land_physical_template defines land_template for zero, one, or more Padstack_definition objects.

4.3.886 Padstack_definition to Material_addition_feature_template

Each Padstack_definition has addition_template defined by one or more Material_addition_feature_template objects. Each Material_addition_feature_template defines addition_template for zero, one, or more Padstack_definition objects.

4.3.887 Padstack_definition to Passage_technology

Each Padstack_definition has material_stackup defined by exactly one Passage_technology. Each Passage_technology defines material_stackup for zero, one, or more Padstack_definition objects.

4.3.888 Parameter_assignment to Model_parameter

Each Parameter_assignment has assigned_parameter defined by exactly one Model_parameter. Each Model_parameter defines assigned_parameter for zero, one, or more Parameter_assignment objects.

4.3.889 Parameter_assignment to Coordinated_characteristic

Each Parameter_assignment has parameter_value defined by zero or one Coordinated_characteristic objects. Each Coordinated_characteristic defines parameter_value for zero, one, or more Parameter_assignment objects. This assertion is established through a SELECT type.

4.3.890 Parameter_assignment to Characteristic

Each Parameter_assignment has parameter_value defined by zero or one Characteristic objects. Each Characteristic defines parameter_value for zero, one, or more Parameter_assignment objects. This assertion is established through a SELECT type.

4.3.891 Participant_date_and_time_assignment to Date_and_time

Each Participant_date_and_time_assignment has action_date_and_time defined by exactly one Date_and_time. Each Date_and_time defines action_date_and_time for zero, one, or more Participant_date_and_time_assignment objects.

4.3.892 Participant_date_and_time_assignment to Person_and_organization

Each Participant_date_and_time_assignment has action_participant defined by exactly one Person_and_organization. Each Person_and_organization defines action_participant for zero, one, or more Participant_date_and_time_assignment objects.

4.3.893 Part_connected_terminals_definition to Part_terminal

Each Part_connected_terminals_definition has connected_terminals defined by two or more Part_terminal objects. Each Part_terminal defines connected_terminals for zero or one Part_connected_terminals_definition objects.

4.3.894 Part_connected_terminals_definition to Physical_unit_usage_view

Each Part_connected_terminals_definition has associated_definition defined by exactly one Physical_unit_usage_view. Each Physical_unit_usage_view defines associated_definition for zero, one, or more Part_connected_terminals_definition objects.

4.3.895 Part_device to Ee_material

Each Part_device has bonding_agent defined by exactly one Ee_material. Each Ee_material defines bonding_agent for zero, one, or more Part_device objects.

4.3.896 Part_device to Bare_die

Each Part_device has included_device defined by exactly one Bare_die. Each Bare_die defines included_device for zero, one, or more Part_device objects.

4.3.897 Part_device to Packaged_part

Each Part_device has defined_packaged_part defined by exactly one Packaged_part. Each Packaged_part defines defined_packaged_part for zero, one, or more Part_device objects.

4.3.898 Part_device_terminal to Part_device

Each Part_device_terminal has terminated_device defined by exactly one Part_device. Each Part_device defines terminated_device for zero, one, or more Part_device_terminal objects.

4.3.899 Part_device_terminal to Minimally_defined_bare_die_terminal

Each Part_device_terminal has used_bare_die_terminal defined by exactly one Minimally_defined_bare_die_terminal. Each Minimally_defined_bare_die_terminal defines used_bare_die_terminal for zero, one, or more Part_device_terminal objects.

4.3.900 Part_device_terminal_model_port_assignment to Analytical_model_port

Each Part_device_terminal_model_port_assignment has internal_port defined by exactly one Analytical_model_port. Each Analytical_model_port defines internal_port for zero, one, or more Part_device_terminal_model_port_assignment objects.

4.3.901 Part_device_terminal_model_port_assignment to Part_device_terminal

Each Part_device_terminal_model_port_assignment has internal_connection defined by exactly one Part_device_terminal. Each Part_device_terminal defines internal_connection for zero, one, or more Part_device_terminal_model_port_assignment objects.

4.3.902 Part_feature to Part_feature

Each Part_feature has precedent_feature defined by exactly one Part_feature. Each Part_feature defines precedent_feature for zero, one, or more Part_feature objects.

4.3.903 Part_feature to Physical_unit_usage_view

Each Part_feature has associated_definition defined by exactly one Physical_unit_usage_view. Each Physical_unit_usage_view defines associated_definition for zero, one, or more Part_feature objects.

4.3.904 Part_feature_make_from_relationship to Part_feature

Each Part_feature_make_from_relationship has resultant_feature defined by exactly one Part_feature. Each Part_feature defines resultant_feature for zero, one, or more Part_feature_make_from_relationship objects. Each Part_feature_make_from_relationship has reusable_feature defined by exactly one Part_feature. Each Part_feature defines reusable_feature for zero, one, or more Part_feature_make_from_relationship objects.

4.3.905 Part_feature_make_from_relationship to Engineering_make_from

Each Part_feature_make_from_relationship has associated_make_from defined by exactly one Engineering_make_from. Each Engineering_make_from defines associated_make_from for zero, one, or more Part_feature_make_from_relationship objects.

4.3.906 Part_feature_model_port_assignment to Analytical_model_port

Each Part_feature_model_port_assignment has target defined by exactly one Analytical_model_port. Each Analytical_model_port defines target for zero, one, or more Part_feature_model_port_assignment objects.

4.3.907 Part_feature_model_port_assignment to Part_feature

Each Part_feature_model_port_assignment has source defined by exactly one Part_feature. Each Part_feature defines source for zero, one, or more Part_feature_model_port_assignment objects.

4.3.908 Part_feature_placement_link to Part_feature

Each Part_feature_placement_link has subsequent_element defined by exactly one Part_feature. Each Part_feature defines subsequent_element for zero, one, or more Part_feature_placement_link objects. Each Part_feature_placement_link has precedent_element defined by exactly one Part_feature. Each Part_feature defines precedent_element for zero, one, or more Part_feature_placement_link objects.

4.3.909 Part_interface_access_feature to Connection_zone

Each Part_interface_access_feature has interface_access_area defined by one or more Connection_zone objects. Each Connection_zone defines interface_access_area for zero, one, or more Part_interface_access_feature objects.

4.3.910 Part_interface_access_feature to Part_feature

Each Part_interface_access_feature has accessed_interface defined by exactly one Part_feature. Each Part_feature defines accessed_interface for zero, one, or more Part_interface_access_feature objects.

4.3.911 Part_linear_array_feature to Length_data_element

Each Part_linear_array_feature has terminus_node_positive_x_displacement defined by exactly one Length_data_element. Each Length_data_element defines terminus_node_positive_x_displacement for zero, one, or more Part_linear_array_feature objects. Each Part_linear_array_feature has initial_node_positive_x_displacement defined by exactly one Length_data_element. Each Length_data_element defines initial_node_positive_x_displacement for zero, one, or more Part_linear_array_feature objects.

4.3.912 Part_linear_array_feature to Part_feature_placement_link

Each Part_linear_array_feature has element defined by one or more Part_feature_placement_link objects. Each Part_feature_placement_link defines element for zero, one, or more Part_linear_array_feature objects.

4.3.913 Part_linear_array_feature_link to Part_linear_array_feature

Each Part_linear_array_feature_link has subsequent_element defined by exactly one Part_linear_array_feature. Each Part_linear_array_feature defines subsequent_element for zero, one, or more Part_linear_array_feature_link objects. Each Part_linear_array_feature_link has precedent_element defined by exactly one Part_linear_array_feature. Each Part_linear_array_feature defines precedent_element for zero, one, or more Part_linear_array_feature_link objects.

4.3.914 Part_mating_feature to Connection_zone

Each Part_mating_feature has mating_area defined by one or more Connection_zone objects. Each Connection_zone defines mating_area for zero, one, or more Part_mating_feature objects.

4.3.915 Part_mounting_feature to Connection_zone

Each Part_mounting_feature has mounting_area defined by one or more Connection_zone objects. Each Connection_zone defines mounting_area for zero, one, or more Part_mounting_feature objects.

4.3.916 Part_rectangular_array_feature to Length_data_element

Each Part_rectangular_array_feature has terminus_row_positive_y_displacement defined by exactly one Length_data_element. Each Length_data_element defines terminus_row_positive_y_displacement for zero, one, or more Part_rectangular_array_feature objects. Each Part_rectangular_array_feature has initial_row_positive_y_displacement defined by exactly one Length_data_element. Each Length_data_element defines initial_row_positive_y_displacement for zero, one, or more Part_rectangular_array_feature objects.

4.3.917 Part_rectangular_array_feature to Part_linear_array_feature_link

Each Part_rectangular_array_feature has element defined by one or more Part_linear_array_feature_link objects. Each Part_linear_array_feature_link defines element for zero, one, or more Part_rectangular_array_feature objects.

4.3.918 Part_template to Template_definition

Each Part_template has associated_definition defined by exactly one Template_definition. Each Template_definition defines associated_definition for zero, one, or more Part_template objects.

4.3.919 Part_template to Analytical_representation

Each Part_template has analytical_part_template_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines analytical_part_template_representation for zero, one, or more Part_template objects.

4.3.920 Part_template to Coordinated_characteristic

Each Part_template has physical_characteristic defined by zero or one Coordinated_characteristic objects. Each Coordinated_characteristic defines physical_characteristic for zero, one, or more Part_template objects. This assertion is established through a SELECT type.

4.3.921 Part_template to Characteristic

Each Part_template has physical_characteristic defined by zero or one Characteristic objects. Each Characteristic defines physical_characteristic for zero, one, or more Part_template objects. This assertion is established through a SELECT type.

4.3.922 Part_template to Analytical_model

Each Part_template has part_template_analytical_model defined by exactly one Analytical_model. Each Analytical_model defines part_template_analytical_model for zero, one, or more Part_template objects.

4.3.923 Part_template_3d_keepout_shape to Cartesian_point

Each Part_template_3d_keepout_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Part_template_3d_keepout_shape objects.

4.3.924 Part_template_3d_keepout_shape to External_definition

Each Part_template_3d_keepout_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Part_template_3d_keepout_shape objects. This assertion is established through a SELECT type.

4.3.925 Part_template_3d_keepout_shape to Part_template

Each Part_template_3d_keepout_shape has shape_characterized_part_template defined by exactly one Part_template. Each Part_template defines shape_characterized_part_template for zero, one, or more Part_template_3d_keepout_shape objects.

4.3.926 Part_template_3d_keepout_shape to Ee_requirement_occurrence

Each Part_template_3d_keepout_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Part_template_3d_keepout_shape objects.

4.3.927 Part_template_3d_shape to Part_template

Each Part_template_3d_shape has shape_characterized_part_template defined by exactly one Part_template. Each Part_template defines shape_characterized_part_template for zero, one, or more Part_template_3d_shape objects.

4.3.928 Part_template_non_planar_2d_shape to Part_template

Each Part_template_non_planar_2d_shape has shape_characterized_part_template defined by exactly one Part_template. Each Part_template defines shape_characterized_part_template for zero, one, or more Part_template_non_planar_2d_shape objects.

4.3.929 Part_template_non_planar_2d_shape to Axis_placement_3d

Each Part_template_non_planar_2d_shape has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Part_template_non_planar_2d_shape objects.

4.3.930 Part_template_or_physical_unit_2d_shape_representation_relationship to Part_template_planar_shape

Each Part_template_or_physical_unit_2d_shape_representation_relationship has related_part_shape defined by zero or one Part_template_planar_shape objects. Each Part_template_planar_shape defines related_part_shape for zero, one, or more Part_template_or_physical_unit_2d_shape_representation_relationship objects. This assertion is established through a SELECT type. Each Part_template_or_physical_unit_2d_shape_representation_relationship has principal_part_shape defined by zero or one Part_template_planar_shape objects. Each Part_template_planar_shape defines principal_part_shape for zero, one, or more Part_template_or_physical_unit_2d_shape_representation_relationship objects. This assertion is established through a SELECT type.

4.3.931 Part_template_or_physical_unit_2d_shape_representation_relationship to Physical_unit_planar_shape

Each Part_template_or_physical_unit_2d_shape_representation_relationship has related_part_shape defined by zero or one Physical_unit_planar_shape objects. Each Physical_unit_planar_shape defines related_part_shape for zero, one, or more Part_template_or_physical_unit_2d_shape_representation_relationship objects. This assertion is established through a SELECT type. Each Part_template_or_physical_unit_2d_shape_representation_relationship has principal_part_shape defined by zero or one Physical_unit_planar_shape objects. Each Physical_unit_planar_shape defines principal_part_shape for zero, one, or more Part_template_or_physical_unit_2d_shape_representation_relationship objects. This assertion is established through a SELECT type.

4.3.932 Part_template_or_physical_unit_3d_shape_representation_relationship to Part_template_3d_shape

Each Part_template_or_physical_unit_3d_shape_representation_relationship has related_part_shape defined by zero or one Part_template_3d_shape objects. Each Part_template_3d_shape defines related_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type. Each Part_template_or_physical_unit_3d_shape_representation_relationship has principal_part_shape defined by zero or one Part_template_3d_shape objects. Each Part_template_3d_shape defines principal_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type.

4.3.933 Part_template_or_physical_unit_3d_shape_representation_relationship to Part_template_non_planar_2d_shape

Each Part_template_or_physical_unit_3d_shape_representation_relationship has related_part_shape defined by zero or one Part_template_non_planar_2d_shape objects. Each Part_template_non_planar_2d_shape defines related_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type. Each Part_template_or_physical_unit_3d_shape_representation_relationship has principal_part_shape defined by zero or one Part_template_non_planar_2d_shape objects. Each Part_template_non_planar_2d_shape defines principal_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type.

4.3.934 Part_template_or_physical_unit_3d_shape_representation_relationship to Physical_unit_3d_shape

Each Part_template_or_physical_unit_3d_shape_representation_relationship has related_part_shape defined by zero or one Physical_unit_3d_shape objects. Each Physical_unit_3d_shape defines related_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type. Each Part_template_or_physical_unit_3d_shape_representation_relationship has principal_part_shape defined by zero or one Physical_unit_3d_shape objects. Each Physical_unit_3d_shape defines principal_part_shape for zero, one, or more Part_template_or_physical_unit_3d_shape_representation_relationship objects. This assertion is established through a SELECT type.

4.3.935 Part_template_planar_keepout_shape to External_definition

Each Part_template_planar_keepout_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Part_template_planar_keepout_shape objects. This assertion is established through a SELECT type.

4.3.936 Part_template_planar_keepout_shape to Cartesian_point

Each Part_template_planar_keepout_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Part_template_planar_keepout_shape objects.

4.3.937 Part_template_planar_keepout_shape to Part_template

Each Part_template_planar_keepout_shape has shape_characterized_part_template defined by exactly one Part_template. Each Part_template defines shape_characterized_part_template for zero, one, or more Part_template_planar_keepout_shape objects.

4.3.938 Part_template_planar_keepout_shape to Ee_requirement_occurrence

Each Part_template_planar_keepout_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Part_template_planar_keepout_shape objects.

4.3.939 Part_template_planar_shape to Part_template

Each Part_template_planar_shape has shape_characterized_part_template defined by exactly one Part_template. Each Part_template defines shape_characterized_part_template for zero, one, or more Part_template_planar_shape objects.

4.3.940 Part_terminal_external_reference to Part_template

Each Part_terminal_external_reference has associated_definition defined by zero or one Part_template objects. Each Part_template defines associated_definition for zero, one, or more Part_terminal_external_reference objects. This assertion is established through a SELECT type.

4.3.941 Part_terminal_external_reference to Physical_unit

Each Part_terminal_external_reference has associated_definition defined by zero or one Physical_unit objects. Each Physical_unit defines associated_definition for zero, one, or more Part_terminal_external_reference objects. This assertion is established through a SELECT type.

4.3.942 Part_text_template to Length_data_element

Each Part_text_template has stroke_width defined by exactly one Length_data_element. Each Length_data_element defines stroke_width for zero, one, or more Part_text_template objects. Each Part_text_template has maximum_font_vertical_extent defined by exactly one Length_data_element. Each Length_data_element defines maximum_font_vertical_extent for zero, one, or more Part_text_template objects. Each Part_text_template has maximum_font_horizontal_extent defined by exactly one Length_data_element. Each Length_data_element defines maximum_font_horizontal_extent for zero, one, or more Part_text_template objects.

4.3.943 Part_text_template to Angle_data_element

Each Part_text_template has slant_angle defined by exactly one Angle_data_element. Each Angle_data_element defines slant_angle for zero, one, or more Part_text_template objects.

4.3.944 Part_text_template to Registered_font

Each Part_text_template has font defined by zero or one Registered_font objects. Each Registered_font defines font for zero, one, or more Part_text_template objects. This assertion is established through a SELECT type.

4.3.945 Part_text_template to Font_definition

Each Part_text_template has font defined by zero or one Font_definition objects. Each Font_definition defines font for zero, one, or more Part_text_template objects. This assertion is established through a SELECT type.

4.3.946 Passage_technology to Process_specification

Each Passage_technology has specification defined by exactly one Process_specification. Each Process_specification defines specification for zero, one, or more Passage_technology objects.

4.3.947 Passage_technology to Length_data_element

Each Passage_technology has minimum_as_finished_deposition_thickness defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_deposition_thickness for zero, one, or more Passage_technology objects. Each Passage_technology has maximum_as_finished_deposition_thickness defined by exactly one Length_data_element. Each Length_data_element defines maximum_as_finished_deposition_thickness for zero, one, or more Passage_technology objects. Each Passage_technology has minimum_as_finished_passage_extent defined by exactly one Length_data_element. Each Length_data_element defines minimum_as_finished_passage_extent for zero, one, or more Passage_technology objects. Each Passage_technology has maximum_as_finished_passage_extent defined by exactly one Length_data_element. Each Length_data_element defines maximum_as_finished_passage_extent for zero, one, or more Passage_technology objects.

4.3.948 Passage_technology to Ee_material

Each Passage_technology has deposition_material defined by exactly one Ee_material. Each Ee_material defines deposition_material for zero, one, or more Passage_technology objects.

4.3.949 Passage_technology to Inter_stratum_extent

Each Passage_technology has as_finished_inter_stratum_extent defined by exactly one Inter_stratum_extent. Each Inter_stratum_extent defines as_finished_inter_stratum_extent for zero, one, or more Passage_technology objects.

4.3.950 Passage_technology to Interconnect_module

Each Passage_technology has associated_definition defined by exactly one Interconnect_module. Each Interconnect_module defines associated_definition for zero, one, or more Passage_technology objects.

4.3.951 Pca to Pca_usage_view

Each Pca has usage_view defined by exactly one Pca_usage_view. Each Pca_usage_view defines usage_view for zero, one, or more Pca objects.

4.3.952 Pca_terminal to Pca_usage_view

Each Pca_terminal has associated_definition defined by exactly one Pca_usage_view. Each Pca_usage_view defines associated_definition for zero, one, or more Pca_terminal objects.

4.3.953 Pcb to Pcb_usage_view

Each Pcb has usage_view defined by exactly one Pcb_usage_view. Each Pcb_usage_view defines usage_view for zero, one, or more Pcb objects.

4.3.954 Person_and_organization to Organization

Each Person_and_organization has assigned_organization defined by exactly one Organization. Each Organization defines assigned_organization for zero, one, or more Person_and_organization objects.

4.3.955 Person_and_organization to Person

Each Person_and_organization has assigned_person defined by exactly one Person. Each Person defines assigned_person for zero, one, or more Person_and_organization objects.

4.3.956 Physical_component to Physical_unit

Each Physical_component has usage_definition defined by exactly one Physical_unit. Each Physical_unit defines usage_definition for zero, one, or more Physical_component objects.

4.3.957 Physical_connectivity_abstraction_map to Conductive_interconnect_element

Each Physical_connectivity_abstraction_map has geometric_context_based_definition defined by exactly one Conductive_interconnect_element. Each Conductive_interconnect_element defines geometric_context_based_definition for zero, one, or more Physical_connectivity_abstraction_map objects.

4.3.958 Physical_connectivity_abstraction_map to Physical_connectivity_element

Each Physical_connectivity_abstraction_map has name_space_context_based_definition defined by exactly one Physical_connectivity_element. Each Physical_connectivity_element defines name_space_context_based_definition for zero, one, or more Physical_connectivity_abstraction_map objects.

4.3.959 Physical_connectivity_definition to Physical_component_terminal

Each Physical_connectivity_definition has associated_terminals defined by two or more Physical_component_terminal objects. Each Physical_component_terminal defines associated_terminals for zero or one Physical_connectivity_definition objects.

4.3.960 Physical_connectivity_definition to Physical_unit_network_definition

Each Physical_connectivity_definition has associated_definition defined by exactly one Physical_unit_network_definition. Each Physical_unit_network_definition defines associated_definition for zero, one, or more Physical_connectivity_definition objects.

4.3.961 Physical_connectivity_definition_relationship to Physical_connectivity_definition

Each Physical_connectivity_definition_relationship has current_definition defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines current_definition for zero, one, or more Physical_connectivity_definition_relationship objects. Each Physical_connectivity_definition_relationship has previous_definition defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines previous_definition for zero, one, or more Physical_connectivity_definition_relationship objects.

4.3.962 Physical_connectivity_definition_relationship to Ee_requirement_occurrence

Each Physical_connectivity_definition_relationship has requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines requirement for zero, one, or more Physical_connectivity_definition_relationship objects.

4.3.963 Physical_connectivity_element to Physical_connectivity_structure_definition

Each Physical_connectivity_element has connectivity_context defined by exactly one Physical_connectivity_structure_definition. Each Physical_connectivity_structure_definition defines connectivity_context for one or more Physical_connectivity_element objects.

4.3.964 Physical_connectivity_element to Component_terminal

Each Physical_connectivity_element has end_terminus defined by zero or one Component_terminal objects. Each Component_terminal defines end_terminus for zero, one, or more Physical_connectivity_element objects. This assertion is established through a SELECT type. Each Physical_connectivity_element has start_terminus defined by zero or one Component_terminal objects. Each Component_terminal defines start_terminus for zero, one, or more Physical_connectivity_element objects. This assertion is established through a SELECT type.

4.3.965 Physical_connectivity_element to Topological_junction

Each Physical_connectivity_element has end_terminus defined by zero or one Topological_junction objects. Each Topological_junction defines end_terminus for zero, one, or more Physical_connectivity_element objects. This assertion is established through a SELECT type. Each Physical_connectivity_element has start_terminus defined by zero or one Topological_junction objects. Each Topological_junction defines start_terminus for zero, one, or more Physical_connectivity_element objects. This assertion is established through a SELECT type.

4.3.966 Physical_connectivity_interrupting_cutout to Conductive_interconnect_element

Each Physical_connectivity_interrupting_cutout has interrupted_connectivity_element defined by one or more Conductive_interconnect_element objects. Each Conductive_interconnect_element defines interrupted_connectivity_element for zero, one, or more Physical_connectivity_interrupting_cutout objects.

4.3.967 Physical_connectivity_make_from_relationship to Physical_connectivity_definition

Each Physical_connectivity_make_from_relationship has resultant_definition defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines resultant_definition for zero, one, or more Physical_connectivity_make_from_relationship objects. Each Physical_connectivity_make_from_relationship has reusable_definition defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines reusable_definition for zero, one, or more Physical_connectivity_make_from_relationship objects.

4.3.968 Physical_network to Part_connected_terminals_definition

Each Physical_network has reference_connected_terminals defined by exactly one Part_connected_terminals_definition. Each Part_connected_terminals_definition defines reference_connected_terminals for zero, one, or more Physical_network objects.

4.3.969 Physical_network to Join_relationship

Each Physical_network has interconnect_module_network_topology_element defined by one or more Join_relationship objects. Each Join_relationship defines interconnect_module_network_topology_element for zero, one, or more Physical_network objects.

4.3.970 Physical_network to Laminate_component_join_terminal

Each Physical_network has connectivity_requirement_element defined by two or more Laminate_component_join_terminal objects. Each Laminate_component_join_terminal defines connectivity_requirement_element for zero or one Physical_network objects.

4.3.971 Physical_network to Interconnect_module

Each Physical_network has prior_associated_definition defined by one or more Interconnect_module objects. Each Interconnect_module defines prior_associated_definition for zero, one, or more Physical_network objects. Each Physical_network has current_associated_definition defined by exactly one Interconnect_module. Each Interconnect_module defines current_associated_definition for zero, one, or more Physical_network objects.

4.3.972 Physical_node_requirement_to_implementing_component_allocation to Assembly_joint

Each Physical_node_requirement_to_implementing_component_allocation has implementation_joint_set defined by one or more Assembly_joint objects. Each Assembly_joint defines implementation_joint_set for zero, one, or more Physical_node_requirement_to_implementing_component_allocation objects.

4.3.973 Physical_node_requirement_to_implementing_component_allocation to Interface_mounted_join

Each Physical_node_requirement_to_implementing_component_allocation has implementation_interface_joint_set defined by one or more Interface_mounted_join objects. Each Interface_mounted_join defines implementation_interface_joint_set for zero, one, or more Physical_node_requirement_to_implementing_component_allocation objects.

4.3.974 Physical_node_requirement_to_implementing_component_allocation to Assembly_component

Each Physical_node_requirement_to_implementing_component_allocation has implementation_component defined by exactly one Assembly_component. Each Assembly_component defines implementation_component for zero, one, or more Physical_node_requirement_to_implementing_component_allocation objects.

4.3.975 Physical_node_requirement_to_implementing_component_allocation to Physical_connectivity_definition

Each Physical_node_requirement_to_implementing_component_allocation has physical_node_requirement defined by exactly one Physical_connectivity_definition. Each Physical_connectivity_definition defines physical_node_requirement for zero, one, or more Physical_node_requirement_to_implementing_component_allocation objects.

4.3.976 Physical_unit to Ee_requirement_occurrence

Each Physical_unit has design_requirement defined by one or more Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines design_requirement for zero, one, or more Physical_unit objects.

4.3.977 Physical_unit to Coordinated_characteristic

Each Physical_unit has physical_device_coordinated_characteristic defined by one or more Coordinated_characteristic objects. Each Coordinated_characteristic defines physical_device_coordinated_characteristic for zero, one, or more Physical_unit objects.

4.3.978 Physical_unit to Characteristic

Each Physical_unit has physical_device_characteristic defined by one or more Characteristic objects. Each Characteristic defines physical_device_characteristic for zero, one, or more Physical_unit objects.

4.3.979 Physical_unit to Technology

Each Physical_unit has unit_technology defined by exactly one Technology. Each Technology defines unit_technology for zero, one, or more Physical_unit objects.

4.3.980 Physical_unit to Analytical_representation

Each Physical_unit has analytical_physical_unit_representation defined by one or more Analytical_representation objects. Each Analytical_representation defines analytical_physical_unit_representation for zero, one, or more Physical_unit objects.

4.3.981 Physical_unit_3d_keepout_shape to Cartesian_point

Each Physical_unit_3d_keepout_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Physical_unit_3d_keepout_shape objects.

4.3.982 Physical_unit_3d_keepout_shape to External_definition

Each Physical_unit_3d_keepout_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Physical_unit_3d_keepout_shape objects. This assertion is established through a SELECT type.

4.3.983 Physical_unit_3d_keepout_shape to Physical_unit

Each Physical_unit_3d_keepout_shape has shape_characterized_physical_unit defined by exactly one Physical_unit. Each Physical_unit defines shape_characterized_physical_unit for zero, one, or more Physical_unit_3d_keepout_shape objects.

4.3.984 Physical_unit_3d_keepout_shape to Ee_requirement_occurrence

Each Physical_unit_3d_keepout_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Physical_unit_3d_keepout_shape objects.

4.3.985 Physical_unit_3d_shape to Cartesian_point

Each Physical_unit_3d_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Physical_unit_3d_shape objects.

4.3.986 Physical_unit_3d_shape to External_definition

Each Physical_unit_3d_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Physical_unit_3d_shape objects. This assertion is established through a SELECT type.

4.3.987 Physical_unit_3d_shape to Physical_unit

Each Physical_unit_3d_shape has shape_characterized_physical_unit defined by exactly one Physical_unit. Each Physical_unit defines shape_characterized_physical_unit for zero, one, or more Physical_unit_3d_shape objects.

4.3.988 Physical_unit_3d_shape to Ee_requirement_occurrence

Each Physical_unit_3d_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Physical_unit_3d_shape objects.

4.3.989 Physical_unit_design_view to Physical_unit_usage_view

Each Physical_unit_design_view has usage_view defined by exactly one Physical_unit_usage_view. Each Physical_unit_usage_view defines usage_view for zero, one, or more Physical_unit_design_view objects.

4.3.990 Physical_unit_planar_keepout_shape to External_definition

Each Physical_unit_planar_keepout_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Physical_unit_planar_keepout_shape objects. This assertion is established through a SELECT type.

4.3.991 Physical_unit_planar_keepout_shape to Cartesian_point

Each Physical_unit_planar_keepout_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Physical_unit_planar_keepout_shape objects.

4.3.992 Physical_unit_planar_keepout_shape to Length_data_element

Each Physical_unit_planar_keepout_shape has shape_distance_from_seating_plane defined by exactly one Length_data_element. Each Length_data_element defines shape_distance_from_seating_plane for zero, one, or more Physical_unit_planar_keepout_shape objects.

4.3.993 Physical_unit_planar_keepout_shape to Physical_unit

Each Physical_unit_planar_keepout_shape has shape_characterized_physical_unit defined by exactly one Physical_unit. Each Physical_unit defines shape_characterized_physical_unit for zero, one, or more Physical_unit_planar_keepout_shape objects.

4.3.994 Physical_unit_planar_keepout_shape to Ee_requirement_occurrence

Each Physical_unit_planar_keepout_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Physical_unit_planar_keepout_shape objects.

4.3.995 Physical_unit_planar_shape to External_definition

Each Physical_unit_planar_shape has shape_purpose defined by zero or one External_definition objects. Each External_definition defines shape_purpose for zero, one, or more Physical_unit_planar_shape objects. This assertion is established through a SELECT type.

4.3.996 Physical_unit_planar_shape to Cartesian_point

Each Physical_unit_planar_shape has centroid_location defined by exactly one Cartesian_point. Each Cartesian_point defines centroid_location for zero, one, or more Physical_unit_planar_shape objects.

4.3.997 Physical_unit_planar_shape to Physical_unit

Each Physical_unit_planar_shape has shape_characterized_physical_unit defined by exactly one Physical_unit. Each Physical_unit defines shape_characterized_physical_unit for zero, one, or more Physical_unit_planar_shape objects.

4.3.998 Physical_unit_planar_shape to Ee_requirement_occurrence

Each Physical_unit_planar_shape has application_technology_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines application_technology_constraint for zero, one, or more Physical_unit_planar_shape objects.

4.3.999 Physical_unit_relationship to Physical_unit

Each Physical_unit_relationship has related_unit defined by exactly one Physical_unit. Each Physical_unit defines related_unit for zero, one, or more Physical_unit_relationship objects. Each Physical_unit_relationship has relating_unit defined by exactly one Physical_unit. Each Physical_unit defines relating_unit for zero, one, or more Physical_unit_relationship objects.

4.3.1000 Pitch_class to Range_characteristic

Each Pitch_class has class_member defined by exactly one Range_characteristic. Each Range_characteristic defines class_member for zero, one, or more Pitch_class objects.

4.3.1001 Placement_group to Ee_requirement_occurrence

Each Placement_group has requirement defined by one or more Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines requirement for zero, one, or more Placement_group objects.

4.3.1002 Placement_group_area_assignment to Mounting_restriction_area

Each Placement_group_area_assignment has area defined by exactly one Mounting_restriction_area. Each Mounting_restriction_area defines area for zero, one, or more Placement_group_area_assignment objects.

4.3.1003 Placement_group_area_assignment to Placement_group

Each Placement_group_area_assignment has placed_group defined by exactly one Placement_group. Each Placement_group defines placed_group for zero, one, or more Placement_group_area_assignment objects.

4.3.1004 Placement_group_volume_assignment to Mounting_restriction_volume

Each Placement_group_volume_assignment has volume defined by exactly one Mounting_restriction_volume. Each Mounting_restriction_volume defines volume for zero, one, or more Placement_group_volume_assignment objects.

4.3.1005 Placement_group_volume_assignment to Placement_group

Each Placement_group_volume_assignment has placed_group defined by exactly one Placement_group. Each Placement_group defines placed_group for zero, one, or more Placement_group_volume_assignment objects.

4.3.1006 Planned_characteristic to Coordinated_characteristic

Each Planned_characteristic has planned_coordinated_characteristic defined by exactly one Coordinated_characteristic. Each Coordinated_characteristic defines planned_coordinated_characteristic for zero, one, or more Planned_characteristic objects.

4.3.1007 Planned_characteristic to Assembly_requirement_allocation

Each Planned_characteristic has reference_requirement defined by zero or one Assembly_requirement_allocation objects. Each Assembly_requirement_allocation defines reference_requirement for zero, one, or more Planned_characteristic objects. This assertion is established through a SELECT type.

4.3.1008 Planned_characteristic to Requirement_allocation

Each Planned_characteristic has reference_requirement defined by zero or one Requirement_allocation objects. Each Requirement_allocation defines reference_requirement for zero, one, or more Planned_characteristic objects. This assertion is established through a SELECT type.

4.3.1009 Planned_characteristic to Functional_unit_requirement_allocation

Each Planned_characteristic has reference_requirement defined by zero or one Functional_unit_requirement_allocation objects. Each Functional_unit_requirement_allocation defines reference_requirement for zero, one, or more Planned_characteristic objects. This assertion is established through a SELECT type.

4.3.1010 Planned_date_effectivity to Date_and_time

Each Planned_date_effectivity has end_date defined by zero or one Date_and_time objects. Each Date_and_time defines end_date for zero, one, or more Planned_date_effectivity objects. This assertion is established through a SELECT type. Each Planned_date_effectivity has start_date defined by zero or one Date_and_time objects. Each Date_and_time defines start_date for zero, one, or more Planned_date_effectivity objects. This assertion is established through a SELECT type.

4.3.1011 Planned_date_effectivity to Date

Each Planned_date_effectivity has end_date defined by zero or one Date objects. Each Date defines end_date for zero, one, or more Planned_date_effectivity objects. This assertion is established through a SELECT type. Each Planned_date_effectivity has start_date defined by zero or one Date objects. Each Date defines start_date for zero, one, or more Planned_date_effectivity objects. This assertion is established through a SELECT type.

4.3.1012 Planned_effectivity to Data_element

Each Planned_effectivity has component_quantity defined by exactly one Data_element. Each Data_element defines component_quantity for zero, one, or more Planned_effectivity objects.

4.3.1013 Planned_effectivity to Ee_product_configuration

Each Planned_effectivity has configuration defined by exactly one Ee_product_configuration. Each Ee_product_configuration defines configuration for zero, one, or more Planned_effectivity objects.

4.3.1014 Planned_effectivity to Ee_approval

Each Planned_effectivity has effectivity_approval defined by exactly one Ee_approval. Each Ee_approval defines effectivity_approval for zero, one, or more Planned_effectivity objects.

4.3.1015 Planned_effectivity to Assembly_composition_relationship

Each Planned_effectivity has effective_assembly defined by exactly one Assembly_composition_relationship. Each Assembly_composition_relationship defines effective_assembly for zero, one, or more Planned_effectivity objects.

4.3.1016 Planned_lot_effectivity to Data_element

Each Planned_lot_effectivity has lot_size defined by exactly one Data_element. Each Data_element defines lot_size for zero, one, or more Planned_lot_effectivity objects.

4.3.1017 Plated_cutout_edge_segment to Partially_plated_cutout

Each Plated_cutout_edge_segment has composed_cutout defined by exactly one Partially_plated_cutout. Each Partially_plated_cutout defines composed_cutout for zero, one, or more Plated_cutout_edge_segment objects.

4.3.1018 Plated_interconnect_module_edge_segment to Partially_plated_interconnect_module_edge

Each Plated_interconnect_module_edge_segment has composed_edge defined by exactly one Partially_plated_interconnect_module_edge. Each Partially_plated_interconnect_module_edge defines composed_edge for zero, one, or more Plated_interconnect_module_edge_segment objects.

4.3.1019 Plus_minus_tolerance_characteristic to Numeric_parameter

Each Plus_minus_tolerance_characteristic has component_characteristic defined by exactly three Numeric_parameter objects. Each Numeric_parameter defines component_characteristic for zero, one, or more Plus_minus_tolerance_characteristic objects.

4.3.1020 Plus_minus_tolerance_range_representation to Data_element

Each Plus_minus_tolerance_range_representation has lower_deviation defined by exactly one Data_element. Each Data_element defines lower_deviation for zero, one, or more Plus_minus_tolerance_range_representation objects. Each Plus_minus_tolerance_range_representation has upper_deviation defined by exactly one Data_element. Each Data_element defines upper_deviation for zero, one, or more Plus_minus_tolerance_range_representation objects. Each Plus_minus_tolerance_range_representation has basic_dimension_value defined by exactly one Data_element. Each Data_element defines basic_dimension_value for zero, one, or more Plus_minus_tolerance_range_representation objects.

4.3.1021 Polarity_indication_feature to Package_terminal

Each Polarity_indication_feature has associated_terminal defined by exactly one Package_terminal. Each Package_terminal defines associated_terminal for zero, one, or more Polarity_indication_feature objects.

4.3.1022 Polarity_indication_feature to Package_body_surface

Each Polarity_indication_feature has associated_body_vertical_extent defined by one or two Package_body_surface objects. Each Package_body_surface defines associated_body_vertical_extent for zero, one, or more Polarity_indication_feature objects.

4.3.1023 Polygonal_area to Cartesian_point

Each Polygonal_area has bounds defined by three or more Cartesian_point objects. Each Cartesian_point defines bounds for zero, one, or more Polygonal_area objects.

4.3.1024 Positional_boundary_and_profile_boundary_member_relationship to Tolerance_zone_boundary_member

Each Positional_boundary_and_profile_boundary_member_relationship has base_profile_tolerance_zone_boundary_member defined by exactly one Tolerance_zone_boundary_member. Each Tolerance_zone_boundary_member defines base_profile_tolerance_zone_boundary_member for zero, one, or more Positional_boundary_and_profile_boundary_member_relationship objects.

4.3.1025 Positional_boundary_and_profile_boundary_member_relationship to Profile_related_positional_boundary

Each Positional_boundary_and_profile_boundary_member_relationship has offset_positional_boundary defined by exactly one Profile_related_positional_boundary. Each Profile_related_positional_boundary defines offset_positional_boundary for zero, one, or more Positional_boundary_and_profile_boundary_member_relationship objects.

4.3.1026 Positional_boundary_definition to Positional_boundary

Each Positional_boundary_definition has defined_positional_boundary defined by exactly one Positional_boundary. Each Positional_boundary defines defined_positional_boundary for zero, one, or more Positional_boundary_definition objects.

4.3.1027 Positional_boundary_member_definition to Dimensional_size

Each Positional_boundary_member_definition has associated_dimension defined by exactly one Dimensional_size. Each Dimensional_size defines associated_dimension for zero, one, or more Positional_boundary_member_definition objects.

4.3.1028 Positional_boundary_member_definition to Length_data_element

Each Positional_boundary_member_definition has tolerance_value defined by exactly one Length_data_element. Each Length_data_element defines tolerance_value for zero, one, or more Positional_boundary_member_definition objects.

4.3.1029 Positional_boundary_member_definition to Positional_boundary_member

Each Positional_boundary_member_definition has defined_positional_boundary_member defined by exactly one Positional_boundary_member. Each Positional_boundary_member defines defined_positional_boundary_member for zero, one, or more Positional_boundary_member_definition objects.

4.3.1030 Positional_boundary_offset to Length_data_element

Each Positional_boundary_offset has tolerance_value defined by exactly one Length_data_element. Each Length_data_element defines tolerance_value for zero, one, or more Positional_boundary_offset objects.

4.3.1031 Positional_boundary_offset to Positional_boundary_and_profile_boundary_member_relationship

Each Positional_boundary_offset has offset_boundary_relationship defined by exactly one Positional_boundary_and_profile_boundary_member_relationship. Each Positional_boundary_and_profile_boundary_member_relationship defines offset_boundary_relationship for zero, one, or more Positional_boundary_offset objects.

4.3.1032 Pre_defined_datum_axis_3d_symbol to Datum_axis

Each Pre_defined_datum_axis_3d_symbol has associated_element defined by exactly one Datum_axis. Each Datum_axis defines associated_element for zero, one, or more Pre_defined_datum_axis_3d_symbol objects.

4.3.1033 Pre_defined_datum_plane_3d_symbol to Datum_plane

Each Pre_defined_datum_plane_3d_symbol has associated_element defined by exactly one Datum_plane. Each Datum_plane defines associated_element for zero, one, or more Pre_defined_datum_plane_3d_symbol objects.

4.3.1034 Pre_defined_parallel_datum_axis_2d_symbol to Datum_axis

Each Pre_defined_parallel_datum_axis_2d_symbol has associated_element defined by exactly one Datum_axis. Each Datum_axis defines associated_element for zero, one, or more Pre_defined_parallel_datum_axis_2d_symbol objects.

4.3.1035 Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship to Pre_defined_datum_axis_3d_symbol

Each Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship has projected_volume defined by exactly one Pre_defined_datum_axis_3d_symbol. Each Pre_defined_datum_axis_3d_symbol defines projected_volume for zero, one, or more Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship objects.

4.3.1036 Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship to Pre_defined_parallel_datum_axis_2d_symbol

Each Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship has surface_intersection_result defined by exactly one Pre_defined_parallel_datum_axis_2d_symbol. Each Pre_defined_parallel_datum_axis_2d_symbol defines surface_intersection_result for zero, one, or more Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship objects.

4.3.1037 Pre_defined_perpendicular_datum_axis_2d_symbol to Datum_axis

Each Pre_defined_perpendicular_datum_axis_2d_symbol has associated_element defined by exactly one Datum_axis. Each Datum_axis defines associated_element for zero, one, or more Pre_defined_perpendicular_datum_axis_2d_symbol objects.

4.3.1038 Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship to Pre_defined_datum_axis_3d_symbol

Each Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship has projected_volume defined by exactly one Pre_defined_datum_axis_3d_symbol. Each Pre_defined_datum_axis_3d_symbol defines projected_volume for zero, one, or more Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship objects.

4.3.1039 Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship to Pre_defined_perpendicular_datum_axis_2d_symbol

Each Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship has surface_intersection_result defined by exactly one Pre_defined_perpendicular_datum_axis_2d_symbol. Each Pre_defined_perpendicular_datum_axis_2d_symbol defines surface_intersection_result for zero, one, or more Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship objects.

4.3.1040 Pre_defined_perpendicular_datum_plane_2d_symbol to Datum_axis

Each Pre_defined_perpendicular_datum_plane_2d_symbol has associated_element defined by exactly one Datum_axis. Each Datum_axis defines associated_element for zero, one, or more Pre_defined_perpendicular_datum_plane_2d_symbol objects.

4.3.1041 Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship to Pre_defined_datum_plane_3d_symbol

Each Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship has projected_volume defined by exactly one Pre_defined_datum_plane_3d_symbol. Each Pre_defined_datum_plane_3d_symbol defines projected_volume for zero, one, or more Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship objects.

4.3.1042 Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship to Pre_defined_perpendicular_datum_plane_2d_symbol

Each Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship has surface_intersection_result defined by exactly one Pre_defined_perpendicular_datum_plane_2d_symbol. Each Pre_defined_perpendicular_datum_plane_2d_symbol defines surface_intersection_result for zero, one, or more Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship objects.

4.3.1043 Primary_orientation_feature to Package_body_surface

Each Primary_orientation_feature has associated_body_vertical_extent defined by one or two Package_body_surface objects. Each Package_body_surface defines associated_body_vertical_extent for zero, one, or more Primary_orientation_feature objects.

4.3.1044 Printed_component to Layer

Each Printed_component has required_material_stack defined by exactly one Layer. Each Layer defines required_material_stack for zero, one, or more Printed_component objects.

4.3.1045 Printed_component to Printed_component_link

Each Printed_component has printed_component_stack defined by one or more Printed_component_link objects. Each Printed_component_link defines printed_component_stack for zero, one, or more Printed_component objects.

4.3.1046 Printed_component to Printed_part_template

Each Printed_component has usage_definition defined by exactly one Printed_part_template. Each Printed_part_template defines usage_definition for zero, one, or more Printed_component objects.

4.3.1047 Printed_component to Stratum_feature

Each Printed_component has stratum_feature_implementation defined by one or more Stratum_feature objects. Each Stratum_feature defines stratum_feature_implementation for zero, one, or more Printed_component objects.

4.3.1048 Printed_component_join_terminal to Stratum_feature

Each Printed_component_join_terminal has stratum_concept_implementation defined by exactly one Stratum_feature. Each Stratum_feature defines stratum_concept_implementation for zero, one, or more Printed_component_join_terminal objects.

4.3.1049 Printed_component_join_terminal to Printed_component

Each Printed_component_join_terminal has associated_component defined by exactly one Printed_component. Each Printed_component defines associated_component for one or more Printed_component_join_terminal objects.

4.3.1050 Printed_component_join_terminal to Printed_part_template_join_terminal

Each Printed_component_join_terminal has definition defined by exactly one Printed_part_template_join_terminal. Each Printed_part_template_join_terminal defines definition for zero, one, or more Printed_component_join_terminal objects.

4.3.1051 Printed_component_link to Printed_part_template_link

Each Printed_component_link has definition defined by exactly one Printed_part_template_link. Each Printed_part_template_link defines definition for zero, one, or more Printed_component_link objects.

4.3.1052 Printed_component_link to Printed_component

Each Printed_component_link has subsequent_printed_component defined by exactly one Printed_component. Each Printed_component defines subsequent_printed_component for zero or one Printed_component_link objects. Each Printed_component_link has precedent_printed_component defined by exactly one Printed_component. Each Printed_component defines precedent_printed_component for zero or one Printed_component_link objects.

4.3.1053 Printed_connector_component to Printed_connector_template

Each Printed_connector_component has usage_definition defined by exactly one Printed_connector_template. Each Printed_connector_template defines usage_definition for zero, one, or more Printed_connector_component objects.

4.3.1054 Printed_connector_component_interface_terminal to Stratum_feature

Each Printed_connector_component_interface_terminal has stratum_concept_implementation defined by exactly one Stratum_feature. Each Stratum_feature defines stratum_concept_implementation for zero, one, or more Printed_connector_component_interface_terminal objects.

4.3.1055 Printed_connector_component_interface_terminal to Printed_connector_component

Each Printed_connector_component_interface_terminal has associated_component defined by exactly one Printed_connector_component. Each Printed_connector_component defines associated_component for one or more Printed_connector_component_interface_terminal objects.

4.3.1056 Printed_connector_component_interface_terminal to Printed_part_template_interface_terminal

Each Printed_connector_component_interface_terminal has definition defined by exactly one Printed_part_template_interface_terminal. Each Printed_part_template_interface_terminal defines definition for zero, one, or more Printed_connector_component_interface_terminal objects.

4.3.1057 Printed_connector_template_terminal_relationship to Printed_connector_template

Each Printed_connector_template_terminal_relationship has connector defined by exactly one Printed_connector_template. Each Printed_connector_template defines connector for one or more Printed_connector_template_terminal_relationship objects.

4.3.1058 Printed_connector_template_terminal_relationship to Printed_part_template_interface_terminal

Each Printed_connector_template_terminal_relationship has interface_terminal defined by exactly one Printed_part_template_interface_terminal. Each Printed_part_template_interface_terminal defines interface_terminal for one or more Printed_connector_template_terminal_relationship objects.

4.3.1059 Printed_connector_template_terminal_relationship to Printed_part_template_join_terminal

Each Printed_connector_template_terminal_relationship has join_terminal defined by exactly one Printed_part_template_join_terminal. Each Printed_part_template_join_terminal defines join_terminal for zero or one Printed_connector_template_terminal_relationship objects.

4.3.1060 Printed_part_cross_section_template to Edge_segment_vertex

Each Printed_part_cross_section_template has end_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines end_vertex for zero, one, or more Printed_part_cross_section_template objects. Each Printed_part_cross_section_template has start_vertex defined by exactly one Edge_segment_vertex. Each Edge_segment_vertex defines start_vertex for zero, one, or more Printed_part_cross_section_template objects.

4.3.1061 Printed_part_cross_section_template to Length_data_element

Each Printed_part_cross_section_template has minimum_width defined by exactly one Length_data_element. Each Length_data_element defines minimum_width for zero, one, or more Printed_part_cross_section_template objects. Each Printed_part_cross_section_template has maximum_width defined by exactly one Length_data_element. Each Length_data_element defines maximum_width for zero, one, or more Printed_part_cross_section_template objects.

4.3.1062 Printed_part_cross_section_template to Analytical_representation

Each Printed_part_cross_section_template has minimum_transmission_line_characteristic defined by one or more Analytical_representation objects. Each Analytical_representation defines minimum_transmission_line_characteristic for zero, one, or more Printed_part_cross_section_template objects. Each Printed_part_cross_section_template has maximum_transmission_line_characteristic defined by one or more Analytical_representation objects. Each Analytical_representation defines maximum_transmission_line_characteristic for zero, one, or more Printed_part_cross_section_template objects.

4.3.1063 Printed_part_cross_section_template to Analytical_model

Each Printed_part_cross_section_template has transmission_line_model defined by exactly one Analytical_model. Each Analytical_model defines transmission_line_model for zero, one, or more Printed_part_cross_section_template objects.

4.3.1064 Printed_part_cross_section_template to Printed_part_template_material_link

Each Printed_part_cross_section_template has vertical_material_link defined by one or more Printed_part_template_material_link objects. Each Printed_part_template_material_link defines vertical_material_link for zero, one, or more Printed_part_cross_section_template objects. Each Printed_part_cross_section_template has horizontal_material_link defined by one or more Printed_part_template_material_link objects. Each Printed_part_template_material_link defines horizontal_material_link for zero, one, or more Printed_part_cross_section_template objects.

4.3.1065 Printed_part_cross_section_template_terminal to Printed_part_template_material_link

Each Printed_part_cross_section_template_terminal has material_to_bottom_of_terminal defined by exactly one Printed_part_template_material_link. Each Printed_part_template_material_link defines material_to_bottom_of_terminal for zero, one, or more Printed_part_cross_section_template_terminal objects. Each Printed_part_cross_section_template_terminal has material_to_top_of_terminal defined by exactly one Printed_part_template_material_link. Each Printed_part_template_material_link defines material_to_top_of_terminal for zero, one, or more Printed_part_cross_section_template_terminal objects. Each Printed_part_cross_section_template_terminal has material_to_right_of_terminal defined by exactly one Printed_part_template_material_link. Each Printed_part_template_material_link defines material_to_right_of_terminal for zero, one, or more Printed_part_cross_section_template_terminal objects. Each Printed_part_cross_section_template_terminal has material_to_left_of_terminal defined by exactly one Printed_part_template_material_link. Each Printed_part_template_material_link defines material_to_left_of_terminal for zero, one, or more Printed_part_cross_section_template_terminal objects.

4.3.1066 Printed_part_cross_section_template_terminal to Printed_part_cross_section_template

Each Printed_part_cross_section_template_terminal has cross_section_definition defined by exactly one Printed_part_cross_section_template. Each Printed_part_cross_section_template defines cross_section_definition for one or more Printed_part_cross_section_template_terminal objects.

4.3.1067 Printed_part_template to Functional_unit_usage_view

Each Printed_part_template has implemented_function defined by exactly one Functional_unit_usage_view. Each Functional_unit_usage_view defines implemented_function for zero, one, or more Printed_part_template objects.

4.3.1068 Printed_part_template to Printed_part_template_link

Each Printed_part_template has printed_part_template_stack defined by one or more Printed_part_template_link objects. Each Printed_part_template_link defines printed_part_template_stack for zero, one, or more Printed_part_template objects.

4.3.1069 Printed_part_template to Stratum_technology_link

Each Printed_part_template has material_stack defined by one or more Stratum_technology_link objects. Each Stratum_technology_link defines material_stack for zero, one, or more Printed_part_template objects.

4.3.1070 Printed_part_template to Stratum_technology

Each Printed_part_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Printed_part_template objects.

4.3.1071 Printed_part_template_connected_terminals_definition to Printed_part_template_terminal

Each Printed_part_template_connected_terminals_definition has connected_terminals defined by two or more Printed_part_template_terminal objects. Each Printed_part_template_terminal defines connected_terminals for zero or one Printed_part_template_connected_terminals_definition objects.

4.3.1072 Printed_part_template_interface_terminal to Printed_connector_template

Each Printed_part_template_interface_terminal has associated_definition defined by exactly one Printed_connector_template. Each Printed_connector_template defines associated_definition for zero, one, or more Printed_part_template_interface_terminal objects.

4.3.1073 Printed_part_template_link to Printed_part_template

Each Printed_part_template_link has subsequent_printed_part_template defined by exactly one Printed_part_template. Each Printed_part_template defines subsequent_printed_part_template for zero, one, or more Printed_part_template_link objects. Each Printed_part_template_link has precedent_printed_part_template defined by exactly one Printed_part_template. Each Printed_part_template defines precedent_printed_part_template for zero, one, or more Printed_part_template_link objects.

4.3.1074 Printed_part_template_material to Stratum_technology

Each Printed_part_template_material has required_material defined by exactly one Stratum_technology. Each Stratum_technology defines required_material for zero, one, or more Printed_part_template_material objects.

4.3.1075 Printed_part_template_material to Length_data_element

Each Printed_part_template_material has minimum_width defined by exactly one Length_data_element. Each Length_data_element defines minimum_width for zero, one, or more Printed_part_template_material objects. Each Printed_part_template_material has maximum_width defined by exactly one Length_data_element. Each Length_data_element defines maximum_width for zero, one, or more Printed_part_template_material objects.

4.3.1076 Printed_part_template_material_link to Template_material_cross_section_boundary

Each Printed_part_template_material_link has subsequent_material defined by exactly one Template_material_cross_section_boundary. Each Template_material_cross_section_boundary defines subsequent_material for zero, one, or more Printed_part_template_material_link objects. Each Printed_part_template_material_link has precedent_material defined by exactly one Template_material_cross_section_boundary. Each Template_material_cross_section_boundary defines precedent_material for zero, one, or more Printed_part_template_material_link objects.

4.3.1077 Printed_part_template_terminal to Connection_zone

Each Printed_part_template_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Printed_part_template_terminal objects.

4.3.1078 Printed_part_template_terminal to Printed_part_template

Each Printed_part_template_terminal has associated_definition defined by exactly one Printed_part_template. Each Printed_part_template defines associated_definition for zero, one, or more Printed_part_template_terminal objects.

4.3.1079 Printed_part_template_terminal_relationship to Printed_part_template_terminal

Each Printed_part_template_terminal_relationship has relating_printed_part_terminal defined by exactly one Printed_part_template_terminal. Each Printed_part_template_terminal defines relating_printed_part_terminal for zero, one, or more Printed_part_template_terminal_relationship objects. Each Printed_part_template_terminal_relationship has related_printed_part_terminal defined by exactly one Printed_part_template_terminal. Each Printed_part_template_terminal defines related_printed_part_terminal for zero, one, or more Printed_part_template_terminal_relationship objects.

4.3.1080 Probe_access_area to Connection_zone

Each Probe_access_area has probe_contact_zone defined by exactly one Connection_zone. Each Connection_zone defines probe_contact_zone for zero, one, or more Probe_access_area objects.

4.3.1081 Probe_access_area to Stratum_feature

Each Probe_access_area has probed_layout_item defined by exactly one Stratum_feature. Each Stratum_feature defines probed_layout_item for zero, one, or more Probe_access_area objects.

4.3.1082 Product_association to Ee_specification

Each Product_association has constraint_source defined by exactly one Ee_specification. Each Ee_specification defines constraint_source for zero, one, or more Product_association objects.

4.3.1083 Product_association to Ee_product_version

Each Product_association has related_product defined by one or more Ee_product_version objects. Each Ee_product_version defines related_product for zero, one, or more Product_association objects. Each Product_association has relating_product defined by exactly one Ee_product_version. Each Ee_product_version defines relating_product for zero, one, or more Product_association objects.

4.3.1084 Profile_boundary_definition to Tolerance_zone_opposing_boundary_set

Each Profile_boundary_definition has defined_zone_boundary defined by exactly one Tolerance_zone_opposing_boundary_set. Each Tolerance_zone_opposing_boundary_set defines defined_zone_boundary for zero, one, or more Profile_boundary_definition objects.

4.3.1085 Profile_boundary_definition to True_profile

Each Profile_boundary_definition has basic_profile defined by exactly one True_profile. Each True_profile defines basic_profile for one or more Profile_boundary_definition objects.

4.3.1086 Profile_boundary_definition_with_offsets to Tolerance_zone_boundary_member_offset

Each Profile_boundary_definition_with_offsets has boundary_offsets defined by exactly two Tolerance_zone_boundary_member_offset objects. Each Tolerance_zone_boundary_member_offset defines boundary_offsets for zero, one, or more Profile_boundary_definition_with_offsets objects.

4.3.1087 Profile_boundary_definition_with_offsets to Tolerance_zone_explicit_opposing_boundary_set

Each Profile_boundary_definition_with_offsets has defined_zone_boundary defined by exactly one Tolerance_zone_explicit_opposing_boundary_set. Each Tolerance_zone_explicit_opposing_boundary_set defines defined_zone_boundary for zero, one, or more Profile_boundary_definition_with_offsets objects.

4.3.1088 Profile_related_positional_boundary_definition to Positional_boundary_offset

Each Profile_related_positional_boundary_definition has boundary_offset defined by exactly one Positional_boundary_offset. Each Positional_boundary_offset defines boundary_offset for zero, one, or more Profile_related_positional_boundary_definition objects.

4.3.1089 Profile_related_positional_boundary_definition to Profile_related_positional_boundary

Each Profile_related_positional_boundary_definition has defined_positional_boundary defined by exactly one Profile_related_positional_boundary. Each Profile_related_positional_boundary defines defined_positional_boundary for zero, one, or more Profile_related_positional_boundary_definition objects.

4.3.1090 Projected_tolerance_zone_definition to Projected_zone_height_characteristic

Each Projected_tolerance_zone_definition has projected_zone_height defined by exactly one Projected_zone_height_characteristic. Each Projected_zone_height_characteristic defines projected_zone_height for zero, one, or more Projected_tolerance_zone_definition objects.

4.3.1091 Projected_zone_and_base_relationship to Tolerance_zone

Each Projected_zone_and_base_relationship has projected_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines projected_zone for zero, one, or more Projected_zone_and_base_relationship objects.

4.3.1092 Projected_zone_and_base_relationship to Physical_feature_or_part_template

Each Projected_zone_and_base_relationship has base_physical_feature defined by exactly one Physical_feature_or_part_template. Each Physical_feature_or_part_template defines base_physical_feature for zero, one, or more Projected_zone_and_base_relationship objects.

4.3.1093 Projected_zone_height_characteristic to Length_data_element

Each Projected_zone_height_characteristic has projected_height_value defined by exactly one Length_data_element. Each Length_data_element defines projected_height_value for zero, one, or more Projected_zone_height_characteristic objects.

4.3.1094 Projected_zone_height_characteristic to Projected_zone_and_base_relationship

Each Projected_zone_height_characteristic has defined_height_of defined by exactly one Projected_zone_and_base_relationship. Each Projected_zone_and_base_relationship defines defined_height_of for zero, one, or more Projected_zone_height_characteristic objects.

4.3.1095 Promissory_usage_relationship to Physical_unit_design_view

Each Promissory_usage_relationship has assembly defined by exactly one Physical_unit_design_view. Each Physical_unit_design_view defines assembly for zero, one, or more Promissory_usage_relationship objects.

4.3.1096 Promissory_usage_relationship to Security_classification

Each Promissory_usage_relationship has security_code defined by exactly one Security_classification. Each Security_classification defines security_code for zero, one, or more Promissory_usage_relationship objects.

4.3.1097 Promissory_usage_relationship to Physical_unit

Each Promissory_usage_relationship has component defined by exactly one Physical_unit. Each Physical_unit defines component for zero, one, or more Promissory_usage_relationship objects.

4.3.1098 Radius_edge_feature_shape to Length_data_element

Each Radius_edge_feature_shape has maximum_radius_length defined by exactly one Length_data_element. Each Length_data_element defines maximum_radius_length for zero, one, or more Radius_edge_feature_shape objects. Each Radius_edge_feature_shape has minimum_radius_length defined by exactly one Length_data_element. Each Length_data_element defines minimum_radius_length for zero, one, or more Radius_edge_feature_shape objects.

4.3.1099 Range_characteristic to Numeric_parameter

Each Range_characteristic has component_characteristic defined by exactly two Numeric_parameter objects. Each Numeric_parameter defines component_characteristic for zero, one, or more Range_characteristic objects.

4.3.1100 Rectangular_area to Length_data_element

Each Rectangular_area has y defined by exactly one Length_data_element. Each Length_data_element defines y for zero, one, or more Rectangular_area objects. Each Rectangular_area has x defined by exactly one Length_data_element. Each Length_data_element defines x for zero, one, or more Rectangular_area objects.

4.3.1101 Rectangular_area to Axis_placement_2d

Each Rectangular_area has position defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines position for zero, one, or more Rectangular_area objects.

4.3.1102 Rectangular_array_placement_group_component to Length_data_element

Each Rectangular_array_placement_group_component has terminus_row_positive_y_displacement defined by exactly one Length_data_element. Each Length_data_element defines terminus_row_positive_y_displacement for zero, one, or more Rectangular_array_placement_group_component objects. Each Rectangular_array_placement_group_component has initial_row_positive_y_displacement defined by exactly one Length_data_element. Each Length_data_element defines initial_row_positive_y_displacement for zero, one, or more Rectangular_array_placement_group_component objects.

4.3.1103 Rectangular_array_placement_group_component to Linear_array_placement_group_component_link

Each Rectangular_array_placement_group_component has element defined by one or more Linear_array_placement_group_component_link objects. Each Linear_array_placement_group_component_link defines element for zero, one, or more Rectangular_array_placement_group_component objects.

4.3.1104 Reference_composition_path to Functional_unit

Each Reference_composition_path has reference_functional_unit defined by exactly one Functional_unit. Each Functional_unit defines reference_functional_unit for zero, one, or more Reference_composition_path objects. Each Reference_composition_path has path_element defined by one or more Functional_unit objects. Each Functional_unit defines path_element for zero, one, or more Reference_composition_path objects.

4.3.1105 Reference_functional_unit_assignment_to_part to Coordinated_characteristic

Each Reference_functional_unit_assignment_to_part has of_property defined by one or more Coordinated_characteristic objects. Each Coordinated_characteristic defines of_property for zero, one, or more Reference_functional_unit_assignment_to_part objects.

4.3.1106 Reference_functional_unit_assignment_to_part to Part_template

Each Reference_functional_unit_assignment_to_part has implementation defined by zero or one Part_template objects. Each Part_template defines implementation for zero, one, or more Reference_functional_unit_assignment_to_part objects. This assertion is established through a SELECT type.

4.3.1107 Reference_functional_unit_assignment_to_part to Physical_unit

Each Reference_functional_unit_assignment_to_part has implementation defined by zero or one Physical_unit objects. Each Physical_unit defines implementation for zero, one, or more Reference_functional_unit_assignment_to_part objects. This assertion is established through a SELECT type.

4.3.1108 Reference_functional_unit_assignment_to_part to Reference_composition_path

Each Reference_functional_unit_assignment_to_part has functional_design_definition_path defined by exactly one Reference_composition_path. Each Reference_composition_path defines functional_design_definition_path for zero, one, or more Reference_functional_unit_assignment_to_part objects.

4.3.1109 Registered_font to Ee_document

Each Registered_font has registration_document defined by exactly one Ee_document. Each Ee_document defines registration_document for zero, one, or more Registered_font objects.

4.3.1110 Registered_style to Ee_document

Each Registered_style has registration_document defined by exactly one Ee_document. Each Ee_document defines registration_document for zero, one, or more Registered_style objects.

4.3.1111 Requirement_allocation to Ee_requirement_occurrence

Each Requirement_allocation has allocated_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines allocated_requirement for zero, one, or more Requirement_allocation objects. Each Requirement_allocation has target_product_object defined by zero or one Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1112 Requirement_allocation to Ee_product_configuration

Each Requirement_allocation has target_product_object defined by zero or one Ee_product_configuration objects. Each Ee_product_configuration defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1113 Requirement_allocation to Ee_product_definition

Each Requirement_allocation has target_product_object defined by zero or one Ee_product_definition objects. Each Ee_product_definition defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1114 Requirement_allocation to Ee_product_version

Each Requirement_allocation has target_product_object defined by zero or one Ee_product_version objects. Each Ee_product_version defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1115 Requirement_allocation to Ee_product

Each Requirement_allocation has target_product_object defined by zero or one Ee_product objects. Each Ee_product defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1116 Requirement_allocation to Part_feature

Each Requirement_allocation has target_product_object defined by zero or one Part_feature objects. Each Part_feature defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1117 Requirement_allocation to Physical_connectivity_element

Each Requirement_allocation has target_product_object defined by zero or one Physical_connectivity_element objects. Each Physical_connectivity_element defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1118 Requirement_allocation to Physical_connectivity_definition

Each Requirement_allocation has target_product_object defined by zero or one Physical_connectivity_definition objects. Each Physical_connectivity_definition defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1119 Requirement_allocation to Laminate_component

Each Requirement_allocation has target_product_object defined by zero or one Laminate_component objects. Each Laminate_component defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1120 Requirement_allocation to Inter_stratum_feature

Each Requirement_allocation has target_product_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1121 Requirement_allocation to Stratum_feature

Each Requirement_allocation has target_product_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1122 Requirement_allocation to Stratum

Each Requirement_allocation has target_product_object defined by zero or one Stratum objects. Each Stratum defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1123 Requirement_allocation to Stratum_surface

Each Requirement_allocation has target_product_object defined by zero or one Stratum_surface objects. Each Stratum_surface defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1124 Requirement_allocation to Assembly_component

Each Requirement_allocation has target_product_object defined by zero or one Assembly_component objects. Each Assembly_component defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1125 Requirement_allocation to Component_external_reference

Each Requirement_allocation has target_product_object defined by zero or one Component_external_reference objects. Each Component_external_reference defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1126 Requirement_allocation to Component_feature

Each Requirement_allocation has target_product_object defined by zero or one Component_feature objects. Each Component_feature defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1127 Requirement_allocation to Component_feature_external_reference

Each Requirement_allocation has target_product_object defined by zero or one Component_feature_external_reference objects. Each Component_feature_external_reference defines target_product_object for zero, one, or more Requirement_allocation objects. This assertion is established through a SELECT type.

4.3.1128 Requirement_composition to Requirement_group_occurrence

Each Requirement_composition has composition defined by exactly one Requirement_group_occurrence. Each Requirement_group_occurrence defines composition for zero, one, or more Requirement_composition objects.

4.3.1129 Requirement_composition to Ee_requirement_occurrence

Each Requirement_composition has element defined by one or more Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines element for zero or one Requirement_composition objects.

4.3.1130 Restraint_condition to Datum_feature_usage_in_datum_relationship

Each Restraint_condition has applied_to defined by zero or one Datum_feature_usage_in_datum_relationship objects. Each Datum_feature_usage_in_datum_relationship defines applied_to for zero, one, or more Restraint_condition objects. This assertion is established through a SELECT type.

4.3.1131 Restraint_condition to Physical_feature_or_part_template

Each Restraint_condition has applied_to defined by zero or one Physical_feature_or_part_template objects. Each Physical_feature_or_part_template defines applied_to for zero, one, or more Restraint_condition objects. This assertion is established through a SELECT type.

4.3.1132 Restraint_condition to Ee_requirement_occurrence

Each Restraint_condition has restraint_description defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines restraint_description for zero, one, or more Restraint_condition objects.

4.3.1133 Routed_interconnect_component to Curve

Each Routed_interconnect_component has routed_centreline_shape defined by exactly one Curve. Each Curve defines routed_centreline_shape for zero, one, or more Routed_interconnect_component objects.

4.3.1134 Routed_physical_component to Curve

Each Routed_physical_component has routed_centreline_shape defined by exactly one Curve. Each Curve defines routed_centreline_shape for zero, one, or more Routed_physical_component objects.

4.3.1135 Routed_printed_component to Inter_stratum_extent

Each Routed_printed_component has component_extent defined by exactly one Inter_stratum_extent. Each Inter_stratum_extent defines component_extent for zero, one, or more Routed_printed_component objects.

4.3.1136 Routed_printed_component to Printed_part_cross_section_template

Each Routed_printed_component has usage_definition defined by exactly one Printed_part_cross_section_template. Each Printed_part_cross_section_template defines usage_definition for zero, one, or more Routed_printed_component objects.

4.3.1137 Routed_printed_component to Curve

Each Routed_printed_component has routed_centreline_shape defined by exactly one Curve. Each Curve defines routed_centreline_shape for zero, one, or more Routed_printed_component objects.

4.3.1138 Rule_action to Ee_rule_version

Each Rule_action has subject_rule defined by exactly one Ee_rule_version. Each Ee_rule_version defines subject_rule for one or more Rule_action objects.

4.3.1139 Rule_action to Participant_date_and_time_assignment

Each Rule_action has subject_action_assignment defined by exactly one Participant_date_and_time_assignment. Each Participant_date_and_time_assignment defines subject_action_assignment for zero, one, or more Rule_action objects.

4.3.1140 Rule_based_constraint_occurrence to Ee_rule_version

Each Rule_based_constraint_occurrence has design_constraint defined by exactly one Ee_rule_version. Each Ee_rule_version defines design_constraint for zero, one, or more Rule_based_constraint_occurrence objects.

4.3.1141 Rule_justification to Rule_action

Each Rule_justification has justified_action defined by exactly one Rule_action. Each Rule_action defines justified_action for zero, one, or more Rule_justification objects.

4.3.1142 Rule_modification to Rule_change_request

Each Rule_modification has modification_rationale defined by exactly one Rule_change_request. Each Rule_change_request defines modification_rationale for zero, one, or more Rule_modification objects.

4.3.1143 Rule_priority to Ee_rule_version

Each Rule_priority has prioritized_rule defined by exactly one Ee_rule_version. Each Ee_rule_version defines prioritized_rule for zero, one, or more Rule_priority objects.

4.3.1144 Rule_set to Rule_priority

Each Rule_set has rule_priority_member defined by one or more Rule_priority objects. Each Rule_priority defines rule_priority_member for zero, one, or more Rule_set objects.

4.3.1145 Rule_set_group to Rule_set

Each Rule_set_group has elements defined by two or more Rule_set objects. Each Rule_set defines elements for zero, one, or more Rule_set_group objects.

4.3.1146 Rule_supersedence to Ee_rule_version

Each Rule_supersedence has superseded_rule defined by exactly one Ee_rule_version. Each Ee_rule_version defines superseded_rule for zero, one, or more Rule_supersedence objects.

4.3.1147 Runout_tolerance_zone_definition_with_specified_angle to Runout_zone_orientation

Each Runout_tolerance_zone_definition_with_specified_angle has runout_orientation defined by exactly one Runout_zone_orientation. Each Runout_zone_orientation defines runout_orientation for zero, one, or more Runout_tolerance_zone_definition_with_specified_angle objects.

4.3.1148 Runout_tolerance_zone_orienting_relationship to Directed_axis

Each Runout_tolerance_zone_orienting_relationship has orienting_axis defined by exactly one Directed_axis. Each Directed_axis defines orienting_axis for zero, one, or more Runout_tolerance_zone_orienting_relationship objects.

4.3.1149 Runout_tolerance_zone_orienting_relationship to Tolerance_zone

Each Runout_tolerance_zone_orienting_relationship has oriented_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines oriented_zone for zero, one, or more Runout_tolerance_zone_orienting_relationship objects.

4.3.1150 Runout_zone_orientation to Angle_data_element

Each Runout_zone_orientation has semi_angle_value defined by exactly one Angle_data_element. Each Angle_data_element defines semi_angle_value for zero, one, or more Runout_zone_orientation objects.

4.3.1151 Runout_zone_orientation to Runout_tolerance_zone_orienting_relationship

Each Runout_zone_orientation has oriented_relationship defined by exactly one Runout_tolerance_zone_orienting_relationship. Each Runout_tolerance_zone_orienting_relationship defines oriented_relationship for zero, one, or more Runout_zone_orientation objects.

4.3.1152 Scalar_terminal_definition_link to Scalar_terminal_definition

Each Scalar_terminal_definition_link has subsequent_terminal_definition defined by exactly one Scalar_terminal_definition. Each Scalar_terminal_definition defines subsequent_terminal_definition for zero, one, or more Scalar_terminal_definition_link objects. Each Scalar_terminal_definition_link has precedent_terminal_definition defined by exactly one Scalar_terminal_definition. Each Scalar_terminal_definition defines precedent_terminal_definition for zero, one, or more Scalar_terminal_definition_link objects.

4.3.1153 Secondary_orientation_feature to Package_body_surface

Each Secondary_orientation_feature has associated_body_vertical_extent defined by one or two Package_body_surface objects. Each Package_body_surface defines associated_body_vertical_extent for zero, one, or more Secondary_orientation_feature objects.

4.3.1154 Security_classification to Date_and_time

Each Security_classification has declassification_date defined by zero or one Date_and_time objects. Each Date_and_time defines declassification_date for zero, one, or more Security_classification objects. This assertion is established through a SELECT type. Each Security_classification has classification_date defined by zero or one Date_and_time objects. Each Date_and_time defines classification_date for zero, one, or more Security_classification objects. This assertion is established through a SELECT type.

4.3.1155 Security_classification to Date

Each Security_classification has declassification_date defined by zero or one Date objects. Each Date defines declassification_date for zero, one, or more Security_classification objects. This assertion is established through a SELECT type. Each Security_classification has classification_date defined by zero or one Date objects. Each Date defines classification_date for zero, one, or more Security_classification objects. This assertion is established through a SELECT type.

4.3.1156 Security_classification to Ee_approval

Each Security_classification has security_classification_approval defined by exactly one Ee_approval. Each Ee_approval defines security_classification_approval for zero, one, or more Security_classification objects.

4.3.1157 Security_classification to Person_and_organization

Each Security_classification has classification_officer defined by exactly one Person_and_organization. Each Person_and_organization defines classification_officer for zero, one, or more Security_classification objects.

4.3.1158 Separate_geometric_tolerance_requirement to Geometric_tolerance_with_referenced_datum_system

Each Separate_geometric_tolerance_requirement has separated_geometric_tolerances defined by one or more Geometric_tolerance_with_referenced_datum_system objects. Each Geometric_tolerance_with_referenced_datum_system defines separated_geometric_tolerances for zero, one, or more Separate_geometric_tolerance_requirement objects.

4.3.1159 Set_characteristic to Characteristic

Each Set_characteristic has values defined by one or more Characteristic objects. Each Characteristic defines values for zero, one, or more Set_characteristic objects.

4.3.1160 Shape_and_ee_product_definition_based_constraint_occurrence to Shape_definition

Each Shape_and_ee_product_definition_based_constraint_occurrence has constraining_shape defined by one or more Shape_definition objects. Each Shape_definition defines constraining_shape for zero, one, or more Shape_and_ee_product_definition_based_constraint_occurrence objects.

4.3.1161 Shape_composition_relationship to Axis_placement

Each Shape_composition_relationship has related_feature_shape_placement defined by exactly one Axis_placement. Each Axis_placement defines related_feature_shape_placement for zero, one, or more Shape_composition_relationship objects.

4.3.1162 Shape_definition to Shape_item

Each Shape_definition has element defined by one or more Shape_item objects. Each Shape_item defines element for one or more Shape_definition objects.

4.3.1163 Shape_definition to Cartesian_coordinate_system

Each Shape_definition has geometric_context defined by exactly one Cartesian_coordinate_system. Each Cartesian_coordinate_system defines geometric_context for zero, one, or more Shape_definition objects.

4.3.1164 Shape_element_composing_relationship to Composite_shape_element

Each Shape_element_composing_relationship has relating_shape_element defined by exactly one Composite_shape_element. Each Composite_shape_element defines relating_shape_element for two or more Shape_element_composing_relationship objects.

4.3.1165 Shape_element_deriving_relationship to Derived_shape_element

Each Shape_element_deriving_relationship has relating_shape_element defined by exactly one Derived_shape_element. Each Derived_shape_element defines relating_shape_element for one or more Shape_element_deriving_relationship objects.

4.3.1166 Shape_element_locating_relationship to Shape_element

Each Shape_element_locating_relationship has located_shape_element defined by exactly one Shape_element. Each Shape_element defines located_shape_element for zero, one, or more Shape_element_locating_relationship objects. Each Shape_element_locating_relationship has locating_shape_element defined by exactly one Shape_element. Each Shape_element defines locating_shape_element for zero, one, or more Shape_element_locating_relationship objects.

4.3.1167 Shape_element_relationship to Shape_element

Each Shape_element_relationship has related_shape_element defined by exactly one Shape_element. Each Shape_element defines related_shape_element for zero, one, or more Shape_element_relationship objects. Each Shape_element_relationship has relating_shape_element defined by exactly one Shape_element. Each Shape_element defines relating_shape_element for zero, one, or more Shape_element_relationship objects.

4.3.1168 Shape_formed_terminal to Lead_form_specification

Each Shape_formed_terminal has terminal_form_specification defined by exactly one Lead_form_specification. Each Lead_form_specification defines terminal_form_specification for zero, one, or more Shape_formed_terminal objects.

4.3.1169 Shield to Assembly_component

Each Shield has shielded_item defined by zero or one Assembly_component objects. Each Assembly_component defines shielded_item for zero, one, or more Shield objects. This assertion is established through a SELECT type.

4.3.1170 Shield to Component_feature

Each Shield has shielded_item defined by zero or one Component_feature objects. Each Component_feature defines shielded_item for zero, one, or more Shield objects. This assertion is established through a SELECT type.

4.3.1171 Shield to Physical_connectivity_definition

Each Shield has shielded_item defined by zero or one Physical_connectivity_definition objects. Each Physical_connectivity_definition defines shielded_item for zero, one, or more Shield objects. This assertion is established through a SELECT type.

4.3.1172 Shield_allocation to Electromagnetic_requirement_occurrence

Each Shield_allocation has requirement defined by exactly one Electromagnetic_requirement_occurrence. Each Electromagnetic_requirement_occurrence defines requirement for zero, one, or more Shield_allocation objects.

4.3.1173 Shield_allocation to Shield

Each Shield_allocation has assembly_assignment defined by exactly one Shield. Each Shield defines assembly_assignment for zero, one, or more Shield_allocation objects.

4.3.1174 Signal to External_definition

Each Signal has signal_category defined by exactly one External_definition. Each External_definition defines signal_category for zero, one, or more Signal objects.

4.3.1175 Signal_property to External_definition

Each Signal_property has signal_property_category defined by exactly one External_definition. Each External_definition defines signal_property_category for zero, one, or more Signal_property objects.

4.3.1176 Signal_property to Signal

Each Signal_property has characterized_signal defined by exactly one Signal. Each Signal defines characterized_signal for zero, one, or more Signal_property objects.

4.3.1177 Signal_property to Characteristic_data_table

Each Signal_property has signal_property_representation defined by exactly one Characteristic_data_table. Each Characteristic_data_table defines signal_property_representation for zero, one, or more Signal_property objects.

4.3.1178 Signal_specification to Functional_unit

Each Signal_specification has characterizing_functional_unit defined by exactly one Functional_unit. Each Functional_unit defines characterizing_functional_unit for zero, one, or more Signal_specification objects.

4.3.1179 Signal_specification to Functional_unit_network_definition

Each Signal_specification has characterization_network defined by exactly one Functional_unit_network_definition. Each Functional_unit_network_definition defines characterization_network for zero, one, or more Signal_specification objects.

4.3.1180 Signal_specification to Signal_property

Each Signal_specification has maximum_characteristic defined by exactly one Signal_property. Each Signal_property defines maximum_characteristic for zero, one, or more Signal_specification objects. Each Signal_specification has minimum_characteristic defined by exactly one Signal_property. Each Signal_property defines minimum_characteristic for zero, one, or more Signal_specification objects.

4.3.1181 Signal_specification to Functional_unit_terminal

Each Signal_specification has reference_access_mechanism defined by exactly one Functional_unit_terminal. Each Functional_unit_terminal defines reference_access_mechanism for zero, one, or more Signal_specification objects. Each Signal_specification has measurement_access_mechanism defined by exactly one Functional_unit_terminal. Each Functional_unit_terminal defines measurement_access_mechanism for zero, one, or more Signal_specification objects.

4.3.1182 Signal_specification to Test_specification

Each Signal_specification has measurement_condition defined by exactly one Test_specification. Each Test_specification defines measurement_condition for zero, one, or more Signal_specification objects.

4.3.1183 Simultaneous_geometric_tolerance_requirement to Geometric_tolerance

Each Simultaneous_geometric_tolerance_requirement has grouped_geometric_tolerances defined by two or more Geometric_tolerance objects. Each Geometric_tolerance defines grouped_geometric_tolerances for zero, one, or more Simultaneous_geometric_tolerance_requirement objects.

4.3.1184 Singular_dimension_representation to Data_element

Each Singular_dimension_representation has dimension_value defined by exactly one Data_element. Each Data_element defines dimension_value for zero, one, or more Singular_dimension_representation objects.

4.3.1185 Special_symbol_laminate_component to Stratum_feature

Each Special_symbol_laminate_component has stratum_feature_implementation defined by exactly one Stratum_feature. Each Stratum_feature defines stratum_feature_implementation for zero, one, or more Special_symbol_laminate_component objects.

4.3.1186 Special_symbol_laminate_component to Special_symbol_part_template

Each Special_symbol_laminate_component has usage_definition defined by exactly one Special_symbol_part_template. Each Special_symbol_part_template defines usage_definition for zero, one, or more Special_symbol_laminate_component objects.

4.3.1187 Special_symbol_part_template to Stratum_technology

Each Special_symbol_part_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Special_symbol_part_template objects.

4.3.1188 Specification_allocation to Ee_product_configuration

Each Specification_allocation has target_product_object defined by zero or one Ee_product_configuration objects. Each Ee_product_configuration defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1189 Specification_allocation to Ee_product_definition

Each Specification_allocation has target_product_object defined by zero or one Ee_product_definition objects. Each Ee_product_definition defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1190 Specification_allocation to Ee_product_version

Each Specification_allocation has target_product_object defined by zero or one Ee_product_version objects. Each Ee_product_version defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1191 Specification_allocation to Ee_product

Each Specification_allocation has target_product_object defined by zero or one Ee_product objects. Each Ee_product defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1192 Specification_allocation to Part_feature

Each Specification_allocation has target_product_object defined by zero or one Part_feature objects. Each Part_feature defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1193 Specification_allocation to Physical_connectivity_element

Each Specification_allocation has target_product_object defined by zero or one Physical_connectivity_element objects. Each Physical_connectivity_element defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1194 Specification_allocation to Physical_connectivity_definition

Each Specification_allocation has target_product_object defined by zero or one Physical_connectivity_definition objects. Each Physical_connectivity_definition defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1195 Specification_allocation to Laminate_component

Each Specification_allocation has target_product_object defined by zero or one Laminate_component objects. Each Laminate_component defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1196 Specification_allocation to Inter_stratum_feature

Each Specification_allocation has target_product_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1197 Specification_allocation to Stratum_feature

Each Specification_allocation has target_product_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1198 Specification_allocation to Stratum

Each Specification_allocation has target_product_object defined by zero or one Stratum objects. Each Stratum defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1199 Specification_allocation to Stratum_surface

Each Specification_allocation has target_product_object defined by zero or one Stratum_surface objects. Each Stratum_surface defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1200 Specification_allocation to Ee_requirement_occurrence

Each Specification_allocation has target_product_object defined by zero or one Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1201 Specification_allocation to Assembly_component

Each Specification_allocation has target_product_object defined by zero or one Assembly_component objects. Each Assembly_component defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1202 Specification_allocation to Component_external_reference

Each Specification_allocation has target_product_object defined by zero or one Component_external_reference objects. Each Component_external_reference defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1203 Specification_allocation to Component_feature

Each Specification_allocation has target_product_object defined by zero or one Component_feature objects. Each Component_feature defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1204 Specification_allocation to Component_feature_external_reference

Each Specification_allocation has target_product_object defined by zero or one Component_feature_external_reference objects. Each Component_feature_external_reference defines target_product_object for zero, one, or more Specification_allocation objects. This assertion is established through a SELECT type.

4.3.1205 Specification_allocation to Ee_specification

Each Specification_allocation has allocated_specification defined by exactly one Ee_specification. Each Ee_specification defines allocated_specification for zero, one, or more Specification_allocation objects.

4.3.1206 Statistical_dimensional_tolerance to Ee_requirement_occurrence

Each Statistical_dimensional_tolerance has statistical_control_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines statistical_control_requirement for zero, one, or more Statistical_dimensional_tolerance objects.

4.3.1207 Statistical_dimensional_tolerance to Dimensional_characteristic

Each Statistical_dimensional_tolerance has alternate_dimension defined by exactly one Dimensional_characteristic. Each Dimensional_characteristic defines alternate_dimension for zero, one, or more Statistical_dimensional_tolerance objects.

4.3.1208 Statistical_geometric_tolerance to Ee_requirement_occurrence

Each Statistical_geometric_tolerance has statistical_control_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines statistical_control_requirement for zero, one, or more Statistical_geometric_tolerance objects.

4.3.1209 Statistical_geometric_tolerance to Geometric_tolerance

Each Statistical_geometric_tolerance has alternate_tolerance defined by exactly one Geometric_tolerance. Each Geometric_tolerance defines alternate_tolerance for zero, one, or more Statistical_geometric_tolerance objects.

4.3.1210 Step_edge_feature_shape to Length_data_element

Each Step_edge_feature_shape has minimum_step_depth defined by exactly one Length_data_element. Each Length_data_element defines minimum_step_depth for zero, one, or more Step_edge_feature_shape objects. Each Step_edge_feature_shape has maximum_step_depth defined by exactly one Length_data_element. Each Length_data_element defines maximum_step_depth for zero, one, or more Step_edge_feature_shape objects. Each Step_edge_feature_shape has maximum_step_width defined by exactly one Length_data_element. Each Length_data_element defines maximum_step_width for zero, one, or more Step_edge_feature_shape objects. Each Step_edge_feature_shape has minimum_step_width defined by exactly one Length_data_element. Each Length_data_element defines minimum_step_width for zero, one, or more Step_edge_feature_shape objects.

4.3.1211 Stratum to Stratum_technology

Each Stratum has of_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_technology for zero, one, or more Stratum objects.

4.3.1212 Stratum to External_definition

Each Stratum has stratum_usage defined by exactly one External_definition. Each External_definition defines stratum_usage for zero, one, or more Stratum objects.

4.3.1213 Stratum_3d_shape to Stratum

Each Stratum_3d_shape has shape_characterized_stratum defined by exactly one Stratum. Each Stratum defines shape_characterized_stratum for zero or one Stratum_3d_shape objects.

4.3.1214 Stratum_average_surface to Stratum

Each Stratum_average_surface has of_stratum defined by exactly one Stratum. Each Stratum defines of_stratum for zero or one Stratum_average_surface objects.

4.3.1215 Stratum_average_surface_shape to Stratum_average_surface

Each Stratum_average_surface_shape has of_surface defined by exactly one Stratum_average_surface. Each Stratum_average_surface defines of_surface for zero or one Stratum_average_surface_shape objects.

4.3.1216 Stratum_concept_relationship to Ee_requirement_occurrence

Each Stratum_concept_relationship has technology_requirement defined by one or more Ee_requirement_occurrence objects. Each Ee_requirement_occurrence defines technology_requirement for zero, one, or more Stratum_concept_relationship objects.

4.3.1217 Stratum_concept_relationship to Feature_shape_definition

Each Stratum_concept_relationship has resulting_shape defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines resulting_shape for zero, one, or more Stratum_concept_relationship objects.

4.3.1218 Stratum_concept_to_physical_usage_view_assignment to Part_feature

Each Stratum_concept_to_physical_usage_view_assignment has defined_part_feature defined by exactly one Part_feature. Each Part_feature defines defined_part_feature for zero or one Stratum_concept_to_physical_usage_view_assignment objects.

4.3.1219 Stratum_concept_to_physical_usage_view_assignment to Laminate_component

Each Stratum_concept_to_physical_usage_view_assignment has assigned_design_object defined by zero or one Laminate_component objects. Each Laminate_component defines assigned_design_object for zero, one, or more Stratum_concept_to_physical_usage_view_assignment objects. This assertion is established through a SELECT type.

4.3.1220 Stratum_concept_to_physical_usage_view_assignment to Inter_stratum_feature

Each Stratum_concept_to_physical_usage_view_assignment has assigned_design_object defined by zero or one Inter_stratum_feature objects. Each Inter_stratum_feature defines assigned_design_object for zero, one, or more Stratum_concept_to_physical_usage_view_assignment objects. This assertion is established through a SELECT type.

4.3.1221 Stratum_concept_to_physical_usage_view_assignment to Stratum_feature

Each Stratum_concept_to_physical_usage_view_assignment has assigned_design_object defined by zero or one Stratum_feature objects. Each Stratum_feature defines assigned_design_object for zero, one, or more Stratum_concept_to_physical_usage_view_assignment objects. This assertion is established through a SELECT type.

4.3.1222 Stratum_concept_to_physical_usage_view_assignment to Stratum

Each Stratum_concept_to_physical_usage_view_assignment has assigned_design_object defined by zero or one Stratum objects. Each Stratum defines assigned_design_object for zero, one, or more Stratum_concept_to_physical_usage_view_assignment objects. This assertion is established through a SELECT type.

4.3.1223 Stratum_concept_to_physical_usage_view_assignment to Stratum_surface

Each Stratum_concept_to_physical_usage_view_assignment has assigned_design_object defined by zero or one Stratum_surface objects. Each Stratum_surface defines assigned_design_object for zero, one, or more Stratum_concept_to_physical_usage_view_assignment objects. This assertion is established through a SELECT type.

4.3.1224 Stratum_feature to Stratum

Each Stratum_feature has resident_stratum defined by exactly one Stratum. Each Stratum defines resident_stratum for zero, one, or more Stratum_feature objects.

4.3.1225 Stratum_feature_3d_shape to Stratum_feature

Each Stratum_feature_3d_shape has represented_stratum_feature defined by exactly one Stratum_feature. Each Stratum_feature defines represented_stratum_feature for zero, one, or more Stratum_feature_3d_shape objects.

4.3.1226 Stratum_feature_conductive_join to Stratum_feature_template_component

Each Stratum_feature_conductive_join has relating_concept defined by exactly one Stratum_feature_template_component. Each Stratum_feature_template_component defines relating_concept for zero, one, or more Stratum_feature_conductive_join objects. Each Stratum_feature_conductive_join has related_concept defined by exactly one Stratum_feature_template_component. Each Stratum_feature_template_component defines related_concept for zero, one, or more Stratum_feature_conductive_join objects.

4.3.1227 Stratum_feature_in_stratum_3d_position to Stratum_feature_non_planar_2d_shape

Each Stratum_feature_in_stratum_3d_position has feature_shape defined by zero or one Stratum_feature_non_planar_2d_shape objects. Each Stratum_feature_non_planar_2d_shape defines feature_shape for zero, one, or more Stratum_feature_in_stratum_3d_position objects. This assertion is established through a SELECT type.

4.3.1228 Stratum_feature_in_stratum_3d_position to Stratum_feature_3d_shape

Each Stratum_feature_in_stratum_3d_position has feature_shape defined by zero or one Stratum_feature_3d_shape objects. Each Stratum_feature_3d_shape defines feature_shape for zero, one, or more Stratum_feature_in_stratum_3d_position objects. This assertion is established through a SELECT type.

4.3.1229 Stratum_feature_in_stratum_3d_position to Axis_placement_3d

Each Stratum_feature_in_stratum_3d_position has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Stratum_feature_in_stratum_3d_position objects.

4.3.1230 Stratum_feature_in_stratum_3d_position to Stratum_3d_shape

Each Stratum_feature_in_stratum_3d_position has stratum_shape defined by exactly one Stratum_3d_shape. Each Stratum_3d_shape defines stratum_shape for zero, one, or more Stratum_feature_in_stratum_3d_position objects.

4.3.1231 Stratum_feature_in_stratum_planar_position to Stratum_feature_styled_area_representation

Each Stratum_feature_in_stratum_planar_position has feature_shape defined by zero or one Stratum_feature_styled_area_representation objects. Each Stratum_feature_styled_area_representation defines feature_shape for zero, one, or more Stratum_feature_in_stratum_planar_position objects. This assertion is established through a SELECT type.

4.3.1232 Stratum_feature_in_stratum_planar_position to Stratum_feature_planar_shape

Each Stratum_feature_in_stratum_planar_position has feature_shape defined by zero or one Stratum_feature_planar_shape objects. Each Stratum_feature_planar_shape defines feature_shape for zero, one, or more Stratum_feature_in_stratum_planar_position objects. This assertion is established through a SELECT type.

4.3.1233 Stratum_feature_in_stratum_planar_position to Axis_placement_2d

Each Stratum_feature_in_stratum_planar_position has transformation defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines transformation for zero, one, or more Stratum_feature_in_stratum_planar_position objects.

4.3.1234 Stratum_feature_in_stratum_planar_position to Stratum_planar_shape

Each Stratum_feature_in_stratum_planar_position has stratum_shape defined by exactly one Stratum_planar_shape. Each Stratum_planar_shape defines stratum_shape for zero, one, or more Stratum_feature_in_stratum_planar_position objects.

4.3.1235 Stratum_feature_non_planar_2d_shape to Stratum_feature

Each Stratum_feature_non_planar_2d_shape has represented_stratum_feature defined by exactly one Stratum_feature. Each Stratum_feature defines represented_stratum_feature for zero, one, or more Stratum_feature_non_planar_2d_shape objects.

4.3.1236 Stratum_feature_planar_shape to Stratum_feature

Each Stratum_feature_planar_shape has represented_stratum_feature defined by exactly one Stratum_feature. Each Stratum_feature defines represented_stratum_feature for zero, one, or more Stratum_feature_planar_shape objects.

4.3.1237 Stratum_feature_shape_stratum_average_surface_shape_relationship to Stratum_average_surface_shape

Each Stratum_feature_shape_stratum_average_surface_shape_relationship has of_stratum_average_surface_shape defined by exactly one Stratum_average_surface_shape. Each Stratum_average_surface_shape defines of_stratum_average_surface_shape for zero, one, or more Stratum_feature_shape_stratum_average_surface_shape_relationship objects.

4.3.1238 Stratum_feature_shape_stratum_average_surface_shape_relationship to Stratum_feature_non_planar_2d_shape

Each Stratum_feature_shape_stratum_average_surface_shape_relationship has of_stratum_feature_shape defined by exactly one Stratum_feature_non_planar_2d_shape. Each Stratum_feature_non_planar_2d_shape defines of_stratum_feature_shape for zero, one, or more Stratum_feature_shape_stratum_average_surface_shape_relationship objects.

4.3.1239 Stratum_feature_shape_stratum_surface_shape_relationship to Stratum_surface_shape

Each Stratum_feature_shape_stratum_surface_shape_relationship has of_stratum_surface_shape defined by exactly one Stratum_surface_shape. Each Stratum_surface_shape defines of_stratum_surface_shape for zero, one, or more Stratum_feature_shape_stratum_surface_shape_relationship objects.

4.3.1240 Stratum_feature_shape_stratum_surface_shape_relationship to Stratum_feature_non_planar_2d_shape

Each Stratum_feature_shape_stratum_surface_shape_relationship has of_stratum_feature_shape defined by exactly one Stratum_feature_non_planar_2d_shape. Each Stratum_feature_non_planar_2d_shape defines of_stratum_feature_shape for zero, one, or more Stratum_feature_shape_stratum_surface_shape_relationship objects.

4.3.1241 Stratum_feature_styled_area_representation to Stratum_feature

Each Stratum_feature_styled_area_representation has represented_stratum_feature defined by exactly one Stratum_feature. Each Stratum_feature defines represented_stratum_feature for zero, one, or more Stratum_feature_styled_area_representation objects.

4.3.1242 Stratum_feature_styled_area_representation to Fill_area_occurrence

Each Stratum_feature_styled_area_representation has shape_styled_area_representation defined by zero or one Fill_area_occurrence objects. Each Fill_area_occurrence defines shape_styled_area_representation for zero, one, or more Stratum_feature_styled_area_representation objects. This assertion is established through a SELECT type.

4.3.1243 Stratum_feature_styled_area_representation to Styled_curve_occurrence

Each Stratum_feature_styled_area_representation has shape_styled_area_representation defined by zero or one Styled_curve_occurrence objects. Each Styled_curve_occurrence defines shape_styled_area_representation for zero, one, or more Stratum_feature_styled_area_representation objects. This assertion is established through a SELECT type.

4.3.1244 Stratum_feature_styled_area_representation to Stratum_feature_planar_shape

Each Stratum_feature_styled_area_representation has derived_planar_shape defined by one or more Stratum_feature_planar_shape objects. Each Stratum_feature_planar_shape defines derived_planar_shape for zero, one, or more Stratum_feature_styled_area_representation objects.

4.3.1245 Stratum_feature_template to Stratum_technology

Each Stratum_feature_template has of_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines of_stratum_technology for zero, one, or more Stratum_feature_template objects.

4.3.1246 Stratum_feature_template_component to Padstack_definition

Each Stratum_feature_template_component has reference_padstack defined by exactly one Padstack_definition. Each Padstack_definition defines reference_padstack for zero, one, or more Stratum_feature_template_component objects.

4.3.1247 Stratum_feature_template_component to Stratum_feature

Each Stratum_feature_template_component has stratum_feature_implementation defined by exactly one Stratum_feature. Each Stratum_feature defines stratum_feature_implementation for zero, one, or more Stratum_feature_template_component objects.

4.3.1248 Stratum_feature_template_component to Stratum_feature_template

Each Stratum_feature_template_component has usage_definition defined by exactly one Stratum_feature_template. Each Stratum_feature_template defines usage_definition for zero, one, or more Stratum_feature_template_component objects.

4.3.1249 Stratum_interconnect_module_3d_position to Axis_placement_3d

Each Stratum_interconnect_module_3d_position has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Stratum_interconnect_module_3d_position objects.

4.3.1250 Stratum_interconnect_module_3d_position to Physical_unit_3d_shape

Each Stratum_interconnect_module_3d_position has interconnect_module_shape defined by exactly one Physical_unit_3d_shape. Each Physical_unit_3d_shape defines interconnect_module_shape for zero, one, or more Stratum_interconnect_module_3d_position objects.

4.3.1251 Stratum_interconnect_module_3d_position to Stratum_3d_shape

Each Stratum_interconnect_module_3d_position has stratum_shape defined by exactly one Stratum_3d_shape. Each Stratum_3d_shape defines stratum_shape for zero, one, or more Stratum_interconnect_module_3d_position objects.

4.3.1252 Stratum_interconnect_module_3d_position to Interconnect_module_stratum_assembly_relationship

Each Stratum_interconnect_module_3d_position has stratum_component defined by exactly one Interconnect_module_stratum_assembly_relationship. Each Interconnect_module_stratum_assembly_relationship defines stratum_component for zero or one Stratum_interconnect_module_3d_position objects.

4.3.1253 Stratum_interconnect_module_planar_position to Axis_placement_2d

Each Stratum_interconnect_module_planar_position has transformation defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines transformation for zero, one, or more Stratum_interconnect_module_planar_position objects.

4.3.1254 Stratum_interconnect_module_planar_position to Physical_unit_planar_shape

Each Stratum_interconnect_module_planar_position has interconnect_module_shape defined by exactly one Physical_unit_planar_shape. Each Physical_unit_planar_shape defines interconnect_module_shape for zero, one, or more Stratum_interconnect_module_planar_position objects.

4.3.1255 Stratum_interconnect_module_planar_position to Stratum_planar_shape

Each Stratum_interconnect_module_planar_position has stratum_shape defined by exactly one Stratum_planar_shape. Each Stratum_planar_shape defines stratum_shape for zero, one, or more Stratum_interconnect_module_planar_position objects.

4.3.1256 Stratum_interconnect_module_planar_position to Interconnect_module_stratum_assembly_relationship

Each Stratum_interconnect_module_planar_position has stratum_component defined by exactly one Interconnect_module_stratum_assembly_relationship. Each Interconnect_module_stratum_assembly_relationship defines stratum_component for zero or one Stratum_interconnect_module_planar_position objects.

4.3.1257 Stratum_make_from_relationship to Stratum

Each Stratum_make_from_relationship has resultant_stratum defined by exactly one Stratum. Each Stratum defines resultant_stratum for zero, one, or more Stratum_make_from_relationship objects. Each Stratum_make_from_relationship has reusable_stratum defined by exactly one Stratum. Each Stratum defines reusable_stratum for zero, one, or more Stratum_make_from_relationship objects.

4.3.1258 Stratum_planar_shape to Stratum

Each Stratum_planar_shape has shape_characterized_stratum defined by exactly one Stratum. Each Stratum defines shape_characterized_stratum for zero or one Stratum_planar_shape objects.

4.3.1259 Stratum_surface to Stratum

Each Stratum_surface has of_stratum defined by exactly one Stratum. Each Stratum defines of_stratum for one or two Stratum_surface objects.

4.3.1260 Stratum_surface_in_stratum_3d_position to Stratum_surface_shape

Each Stratum_surface_in_stratum_3d_position has surface_shape defined by zero or one Stratum_surface_shape objects. Each Stratum_surface_shape defines surface_shape for zero, one, or more Stratum_surface_in_stratum_3d_position objects. This assertion is established through a SELECT type.

4.3.1261 Stratum_surface_in_stratum_3d_position to Stratum_average_surface_shape

Each Stratum_surface_in_stratum_3d_position has surface_shape defined by zero or one Stratum_average_surface_shape objects. Each Stratum_average_surface_shape defines surface_shape for zero, one, or more Stratum_surface_in_stratum_3d_position objects. This assertion is established through a SELECT type.

4.3.1262 Stratum_surface_in_stratum_3d_position to Axis_placement_3d

Each Stratum_surface_in_stratum_3d_position has transformation defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines transformation for zero, one, or more Stratum_surface_in_stratum_3d_position objects.

4.3.1263 Stratum_surface_in_stratum_3d_position to Stratum_3d_shape

Each Stratum_surface_in_stratum_3d_position has stratum_shape defined by exactly one Stratum_3d_shape. Each Stratum_3d_shape defines stratum_shape for zero, one, or more Stratum_surface_in_stratum_3d_position objects.

4.3.1264 Stratum_surface_shape to Stratum_surface

Each Stratum_surface_shape has of_surface defined by exactly one Stratum_surface. Each Stratum_surface defines of_surface for zero or one Stratum_surface_shape objects.

4.3.1265 Stratum_technology to Process_specification

Each Stratum_technology has specification defined by exactly one Process_specification. Each Process_specification defines specification for zero, one, or more Stratum_technology objects.

4.3.1266 Stratum_technology to Length_data_element

Each Stratum_technology has maximum_feature_size_requirement defined by exactly one Length_data_element. Each Length_data_element defines maximum_feature_size_requirement for zero, one, or more Stratum_technology objects. Each Stratum_technology has minimum_finished_feature_spacing defined by exactly one Length_data_element. Each Length_data_element defines minimum_finished_feature_spacing for zero, one, or more Stratum_technology objects. Each Stratum_technology has minimum_finished_feature_size defined by exactly one Length_data_element. Each Length_data_element defines minimum_finished_feature_size for zero, one, or more Stratum_technology objects. Each Stratum_technology has minimum_stratum_thickness defined by exactly one Length_data_element. Each Length_data_element defines minimum_stratum_thickness for zero, one, or more Stratum_technology objects. Each Stratum_technology has maximum_stratum_thickness defined by exactly one Length_data_element. Each Length_data_element defines maximum_stratum_thickness for zero, one, or more Stratum_technology objects.

4.3.1267 Stratum_technology to Ee_material

Each Stratum_technology has stratum_material defined by exactly one Ee_material. Each Ee_material defines stratum_material for zero, one, or more Stratum_technology objects.

4.3.1268 Stratum_technology_link to Stratum_technology_link

Each Stratum_technology_link has precedent_stratum_technology_link defined by exactly one Stratum_technology_link. Each Stratum_technology_link defines precedent_stratum_technology_link for zero, one, or more Stratum_technology_link objects.

4.3.1269 Stratum_technology_link to Stratum_technology

Each Stratum_technology_link has subsequent_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines subsequent_stratum_technology for zero, one, or more Stratum_technology_link objects. Each Stratum_technology_link has precedent_stratum_technology defined by exactly one Stratum_technology. Each Stratum_technology defines precedent_stratum_technology for zero, one, or more Stratum_technology_link objects.

4.3.1270 Structured_text to Structured_text

Each Structured_text has structured_text_content defined by one or more Structured_text objects. Each Structured_text defines structured_text_content for zero, one, or more Structured_text objects. Each Structured_text has precedent_structured_text defined by exactly one Structured_text. Each Structured_text defines precedent_structured_text for zero, one, or more Structured_text objects.

4.3.1271 Structured_text to Ee_text

Each Structured_text has plain_text_content defined by exactly one Ee_text. Each Ee_text defines plain_text_content for zero, one, or more Structured_text objects.

4.3.1272 Structured_text to External_definition

Each Structured_text has content_markup defined by exactly one External_definition. Each External_definition defines content_markup for zero, one, or more Structured_text objects.

4.3.1273 Styled_curve_occurrence to Curve_style

Each Styled_curve_occurrence has of_curve_style defined by exactly one Curve_style. Each Curve_style defines of_curve_style for zero, one, or more Styled_curve_occurrence objects.

4.3.1274 Styled_curve_occurrence to Curve

Each Styled_curve_occurrence has styled_curve defined by exactly one Curve. Each Curve defines styled_curve for zero, one, or more Styled_curve_occurrence objects.

4.3.1275 Styled_text to Registered_font

Each Styled_text has text_font defined by zero or one Registered_font objects. Each Registered_font defines text_font for zero, one, or more Styled_text objects. This assertion is established through a SELECT type.

4.3.1276 Styled_text to Font_definition

Each Styled_text has text_font defined by zero or one Font_definition objects. Each Font_definition defines text_font for zero, one, or more Styled_text objects. This assertion is established through a SELECT type.

4.3.1277 Substitute_product to Assembly_composition_relationship

Each Substitute_product has substitutable_component defined by exactly one Assembly_composition_relationship. Each Assembly_composition_relationship defines substitutable_component for zero, one, or more Substitute_product objects.

4.3.1278 Substitute_product to Ee_product

Each Substitute_product has defined_alternate_product defined by one or more Ee_product objects. Each Ee_product defines defined_alternate_product for zero, one, or more Substitute_product objects.

4.3.1279 Sub_assembly_relationship to Assembly_group_component

Each Sub_assembly_relationship has relating_component defined by exactly one Assembly_group_component. Each Assembly_group_component defines relating_component for one or more Sub_assembly_relationship objects.

4.3.1280 Sub_assembly_relationship to Assembly_component

Each Sub_assembly_relationship has related_component defined by exactly one Assembly_component. Each Assembly_component defines related_component for zero, one, or more Sub_assembly_relationship objects.

4.3.1281 Supplied_product_version to Organization

Each Supplied_product_version has supplier defined by exactly one Organization. Each Organization defines supplier for zero, one, or more Supplied_product_version objects.

4.3.1282 Supplied_product_version to Ee_approval

Each Supplied_product_version has supplied_product_version_approval defined by exactly one Ee_approval. Each Ee_approval defines supplied_product_version_approval for zero, one, or more Supplied_product_version objects.

4.3.1283 Supplied_product_version to Ee_product_version

Each Supplied_product_version has supplied_product_version defined by exactly one Ee_product_version. Each Ee_product_version defines supplied_product_version for zero, one, or more Supplied_product_version objects.

4.3.1284 Surface_prepped_terminal to Surface_finish_specification

Each Surface_prepped_terminal has surface_preparation defined by exactly one Surface_finish_specification. Each Surface_finish_specification defines surface_preparation for zero, one, or more Surface_prepped_terminal objects.

4.3.1285 Symmetrical_tolerance_characteristic to Numeric_parameter

Each Symmetrical_tolerance_characteristic has component_characteristic defined by exactly two Numeric_parameter objects. Each Numeric_parameter defines component_characteristic for zero, one, or more Symmetrical_tolerance_characteristic objects.

4.3.1286 Technology to Ee_material

Each Technology has material_product_system defined by exactly one Ee_material. Each Ee_material defines material_product_system for zero, one, or more Technology objects.

4.3.1287 Technology to External_definition

Each Technology has classification defined by exactly one External_definition. Each External_definition defines classification for zero, one, or more Technology objects.

4.3.1288 Technology to Process_specification

Each Technology has process_technology defined by exactly one Process_specification. Each Process_specification defines process_technology for zero, one, or more Technology objects.

4.3.1289 Template_material_cross_section_boundary to Printed_part_template_material

Each Template_material_cross_section_boundary has associated_definition defined by exactly one Printed_part_template_material. Each Printed_part_template_material defines associated_definition for zero, one, two, three, or four Template_material_cross_section_boundary objects.

4.3.1290 Template_stratum_assignment to Stratum

Each Template_stratum_assignment has bound_stratum defined by exactly one Stratum. Each Stratum defines bound_stratum for zero, one, or more Template_stratum_assignment objects.

4.3.1291 Template_stratum_assignment to Stratum_feature_template

Each Template_stratum_assignment has bound_template defined by zero or one Stratum_feature_template objects. Each Stratum_feature_template defines bound_template for zero, one, or more Template_stratum_assignment objects. This assertion is established through a SELECT type.

4.3.1292 Template_stratum_assignment to Material_addition_feature_template

Each Template_stratum_assignment has bound_template defined by zero or one Material_addition_feature_template objects. Each Material_addition_feature_template defines bound_template for zero, one, or more Template_stratum_assignment objects. This assertion is established through a SELECT type.

4.3.1293 Template_stratum_assignment to Material_removal_feature_template

Each Template_stratum_assignment has bound_template defined by zero or one Material_removal_feature_template objects. Each Material_removal_feature_template defines bound_template for zero, one, or more Template_stratum_assignment objects. This assertion is established through a SELECT type.

4.3.1294 Terminal_pair_port_equivalence_relationship to Analytical_model_scalar_port

Each Terminal_pair_port_equivalence_relationship has scalar_ports defined by exactly two Analytical_model_scalar_port objects. Each Analytical_model_scalar_port defines scalar_ports for zero, one, or more Terminal_pair_port_equivalence_relationship objects.

4.3.1295 Terminal_pair_port_equivalence_relationship to Analytical_model_port

Each Terminal_pair_port_equivalence_relationship has equivalent_port defined by exactly one Analytical_model_port. Each Analytical_model_port defines equivalent_port for zero, one, or more Terminal_pair_port_equivalence_relationship objects.

4.3.1296 Terminal_swap_specification to Functional_unit_terminal_node_assignment

Each Terminal_swap_specification has current_design_object defined by exactly one Functional_unit_terminal_node_assignment. Each Functional_unit_terminal_node_assignment defines current_design_object for zero, one, or more Terminal_swap_specification objects. Each Terminal_swap_specification has previous_design_object defined by exactly one Functional_unit_terminal_node_assignment. Each Functional_unit_terminal_node_assignment defines previous_design_object for zero, one, or more Terminal_swap_specification objects.

4.3.1297 Termination_constraint to Ee_requirement_occurrence

Each Termination_constraint has termination_usage_constraint defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines termination_usage_constraint for zero, one, or more Termination_constraint objects.

4.3.1298 Termination_constraint to Mating_connector_termination

Each Termination_constraint has constrained_termination defined by two or more Mating_connector_termination objects. Each Mating_connector_termination defines constrained_termination for zero, one, or more Termination_constraint objects.

4.3.1299 Tertiary_orientation_feature to Package_body_surface

Each Tertiary_orientation_feature has associated_body_vertical_extent defined by one or two Package_body_surface objects. Each Package_body_surface defines associated_body_vertical_extent for zero, one, or more Tertiary_orientation_feature objects.

4.3.1300 Test_requirement_allocation to Functional_usage_view_to_part_terminal_assignment

Each Test_requirement_allocation has functional_specification_based_test_access defined by one or more Functional_usage_view_to_part_terminal_assignment objects. Each Functional_usage_view_to_part_terminal_assignment defines functional_specification_based_test_access for zero, one, or more Test_requirement_allocation objects.

4.3.1301 Test_requirement_allocation to Part_feature_model_port_assignment

Each Test_requirement_allocation has analytical_model_based_test_access defined by one or more Part_feature_model_port_assignment objects. Each Part_feature_model_port_assignment defines analytical_model_based_test_access for zero, one, or more Test_requirement_allocation objects.

4.3.1302 Test_requirement_allocation to Test_requirement_occurrence

Each Test_requirement_allocation has allocated_requirement defined by exactly one Test_requirement_occurrence. Each Test_requirement_occurrence defines allocated_requirement for zero, one, or more Test_requirement_allocation objects.

4.3.1303 Test_requirement_allocation to Ee_product_version

Each Test_requirement_allocation has target_product_object defined by exactly one Ee_product_version. Each Ee_product_version defines target_product_object for zero, one, or more Test_requirement_allocation objects.

4.3.1304 Thermal_isolation_laminate_component to Thermal_isolation_removal_template

Each Thermal_isolation_laminate_component has usage_definition defined by exactly one Thermal_isolation_removal_template. Each Thermal_isolation_removal_template defines usage_definition for zero, one, or more Thermal_isolation_laminate_component objects.

4.3.1305 Thermal_isolation_requirement_assignment to Ee_requirement_occurrence

Each Thermal_isolation_requirement_assignment has angular_orientation_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines angular_orientation_requirement for zero, one, or more Thermal_isolation_requirement_assignment objects. Each Thermal_isolation_requirement_assignment has effective_current_capacity_requirement defined by exactly one Ee_requirement_occurrence. Each Ee_requirement_occurrence defines effective_current_capacity_requirement for zero, one, or more Thermal_isolation_requirement_assignment objects.

4.3.1306 Thermal_isolation_requirement_assignment to Layout_spacing_requirement_occurrence

Each Thermal_isolation_requirement_assignment has thermal_isolation_spacing_requirement defined by exactly one Layout_spacing_requirement_occurrence. Each Layout_spacing_requirement_occurrence defines thermal_isolation_spacing_requirement for zero, one, or more Thermal_isolation_requirement_assignment objects.

4.3.1307 Thermal_isolation_requirement_assignment to Thermal_isolation_removal_structure

Each Thermal_isolation_requirement_assignment has resulting_structure defined by zero or one Thermal_isolation_removal_structure objects. Each Thermal_isolation_removal_structure defines resulting_structure for zero, one, or more Thermal_isolation_requirement_assignment objects. This assertion is established through a SELECT type.

4.3.1308 Thermal_isolation_requirement_assignment to Thermal_isolation_removal_template

Each Thermal_isolation_requirement_assignment has resulting_structure defined by zero or one Thermal_isolation_removal_template objects. Each Thermal_isolation_removal_template defines resulting_structure for zero, one, or more Thermal_isolation_requirement_assignment objects. This assertion is established through a SELECT type.

4.3.1309 Thermal_requirement_allocation to Thermal_requirement_occurrence

Each Thermal_requirement_allocation has requirement defined by exactly one Thermal_requirement_occurrence. Each Thermal_requirement_occurrence defines requirement for zero, one, or more Thermal_requirement_allocation objects.

4.3.1310 Thermal_requirement_allocation to Thermal_component

Each Thermal_requirement_allocation has assembly_assignment defined by exactly one Thermal_component. Each Thermal_component defines assembly_assignment for zero, one, or more Thermal_requirement_allocation objects.

4.3.1311 Tolerance_characteristic to Numeric_parameter

Each Tolerance_characteristic has component_characteristic defined by one or more Numeric_parameter objects. Each Numeric_parameter defines component_characteristic for zero, one, or more Tolerance_characteristic objects.

4.3.1312 Tolerance_specific_restraint_condition to Geometric_tolerance

Each Tolerance_specific_restraint_condition has affected_tolerances defined by zero or one Geometric_tolerance objects. Each Geometric_tolerance defines affected_tolerances for zero, one, or more Tolerance_specific_restraint_condition objects. This assertion is established through a SELECT type.

4.3.1313 Tolerance_specific_restraint_condition to Dimensional_characteristic

Each Tolerance_specific_restraint_condition has affected_tolerances defined by zero or one Dimensional_characteristic objects. Each Dimensional_characteristic defines affected_tolerances for zero, one, or more Tolerance_specific_restraint_condition objects. This assertion is established through a SELECT type.

4.3.1314 Tolerance_zone to Tolerance_zone_boundary

Each Tolerance_zone has zone_boundary defined by exactly one Tolerance_zone_boundary. Each Tolerance_zone_boundary defines zone_boundary for zero, one, or more Tolerance_zone objects.

4.3.1315 Tolerance_zone_and_viewing_plane_orienting_relationship to Tolerance_zone

Each Tolerance_zone_and_viewing_plane_orienting_relationship has oriented_tolerance_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines oriented_tolerance_zone for zero, one, or more Tolerance_zone_and_viewing_plane_orienting_relationship objects.

4.3.1316 Tolerance_zone_and_viewing_plane_orienting_relationship to Viewing_plane

Each Tolerance_zone_and_viewing_plane_orienting_relationship has orienting_viewing_plane defined by exactly one Viewing_plane. Each Viewing_plane defines orienting_viewing_plane for zero, one, or more Tolerance_zone_and_viewing_plane_orienting_relationship objects.

4.3.1317 Tolerance_zone_boundary_definition to Tolerance_zone_boundary

Each Tolerance_zone_boundary_definition has defined_zone_boundary defined by exactly one Tolerance_zone_boundary. Each Tolerance_zone_boundary defines defined_zone_boundary for zero, one, or more Tolerance_zone_boundary_definition objects.

4.3.1318 Tolerance_zone_boundary_member_offset to Length_data_element

Each Tolerance_zone_boundary_member_offset has offset_value defined by exactly one Length_data_element. Each Length_data_element defines offset_value for zero, one, or more Tolerance_zone_boundary_member_offset objects.

4.3.1319 Tolerance_zone_boundary_member_offset to Tolerance_zone_boundary_member

Each Tolerance_zone_boundary_member_offset has offset_tolerance_zone_boundary_member defined by exactly one Tolerance_zone_boundary_member. Each Tolerance_zone_boundary_member defines offset_tolerance_zone_boundary_member for zero, one, or more Tolerance_zone_boundary_member_offset objects.

4.3.1320 Tolerance_zone_definition to Tolerance_zone_form

Each Tolerance_zone_definition has form defined by exactly one Tolerance_zone_form. Each Tolerance_zone_form defines form for zero, one, or more Tolerance_zone_definition objects.

4.3.1321 Tolerance_zone_definition to Tolerance_zone_boundary_definition

Each Tolerance_zone_definition has boundary_definition defined by exactly one Tolerance_zone_boundary_definition. Each Tolerance_zone_boundary_definition defines boundary_definition for zero, one, or more Tolerance_zone_definition objects.

4.3.1322 Tolerance_zone_definition to Tolerance_zone

Each Tolerance_zone_definition has defined_tolerance_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines defined_tolerance_zone for zero, one, or more Tolerance_zone_definition objects.

4.3.1323 Tolerance_zone_definition_with_per_unit_size_specification to Tolerance_zone_per_unit_size_specification

Each Tolerance_zone_definition_with_per_unit_size_specification has per_unit_size_specification defined by exactly one Tolerance_zone_per_unit_size_specification. Each Tolerance_zone_per_unit_size_specification defines per_unit_size_specification for zero, one, or more Tolerance_zone_definition_with_per_unit_size_specification objects.

4.3.1324 Tolerance_zone_definition_with_specified_orientation to Datum_axis_related_orientation

Each Tolerance_zone_definition_with_specified_orientation has zone_orientation defined by zero or one Datum_axis_related_orientation objects. Each Datum_axis_related_orientation defines zone_orientation for zero, one, or more Tolerance_zone_definition_with_specified_orientation objects. This assertion is established through a SELECT type.

4.3.1325 Tolerance_zone_definition_with_specified_orientation to Viewing_plane_based_orientation

Each Tolerance_zone_definition_with_specified_orientation has zone_orientation defined by zero or one Viewing_plane_based_orientation objects. Each Viewing_plane_based_orientation defines zone_orientation for zero, one, or more Tolerance_zone_definition_with_specified_orientation objects. This assertion is established through a SELECT type.

4.3.1326 Tolerance_zone_definition_with_specified_orientation to Datum_based_vector_orientation

Each Tolerance_zone_definition_with_specified_orientation has zone_orientation defined by zero or one Datum_based_vector_orientation objects. Each Datum_based_vector_orientation defines zone_orientation for zero, one, or more Tolerance_zone_definition_with_specified_orientation objects. This assertion is established through a SELECT type.

4.3.1327 Tolerance_zone_explicit_opposing_boundary_set to Tolerance_zone_boundary_member

Each Tolerance_zone_explicit_opposing_boundary_set has boundary_members defined by exactly two Tolerance_zone_boundary_member objects. Each Tolerance_zone_boundary_member defines boundary_members for zero, one, or more Tolerance_zone_explicit_opposing_boundary_set objects.

4.3.1328 Tolerance_zone_form to Tolerance_zone

Each Tolerance_zone_form has characterized_tolerance_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines characterized_tolerance_zone for zero, one, or more Tolerance_zone_form objects.

4.3.1329 Tolerance_zone_per_unit_size_specification to Length_data_element

Each Tolerance_zone_per_unit_size_specification has per_unit_size defined by one or two Length_data_element objects. Each Length_data_element defines per_unit_size for zero, one, or more Tolerance_zone_per_unit_size_specification objects.

4.3.1330 Tolerance_zone_per_unit_size_specification to Tolerance_zone

Each Tolerance_zone_per_unit_size_specification has specified_tolerance_zone defined by exactly one Tolerance_zone. Each Tolerance_zone defines specified_tolerance_zone for zero, one, or more Tolerance_zone_per_unit_size_specification objects.

4.3.1331 Topological_junction to Physical_connectivity_structure_definition

Each Topological_junction has scope defined by exactly one Physical_connectivity_structure_definition. Each Physical_connectivity_structure_definition defines scope for zero, one, or more Topological_junction objects.

4.3.1332 Trace_template to Curve_style

Each Trace_template has trace_style defined by exactly one Curve_style. Each Curve_style defines trace_style for zero, one, or more Trace_template objects.

4.3.1333 Transmission_line_functional_unit_terminal_allocation to Transmission_line_functional_unit_to_routed_printed_component_allocation

Each Transmission_line_functional_unit_terminal_allocation has functional_unit_design_path defined by exactly one Transmission_line_functional_unit_to_routed_printed_component_allocation. Each Transmission_line_functional_unit_to_routed_printed_component_allocation defines functional_unit_design_path for zero, one, or more Transmission_line_functional_unit_terminal_allocation objects.

4.3.1334 Transmission_line_functional_unit_terminal_allocation to Functional_unit_terminal

Each Transmission_line_functional_unit_terminal_allocation has transmission_line_functional_unit_terminal defined by exactly one Functional_unit_terminal. Each Functional_unit_terminal defines transmission_line_functional_unit_terminal for zero, one, or more Transmission_line_functional_unit_terminal_allocation objects.

4.3.1335 Transmission_line_functional_unit_terminal_allocation to Printed_component_join_terminal

Each Transmission_line_functional_unit_terminal_allocation has implementation defined by exactly one Printed_component_join_terminal. Each Printed_component_join_terminal defines implementation for zero, one, or more Transmission_line_functional_unit_terminal_allocation objects.

4.3.1336 Transmission_line_functional_unit_to_routed_printed_component_allocation to Routed_printed_component

Each Transmission_line_functional_unit_to_routed_printed_component_allocation has implementation defined by exactly one Routed_printed_component. Each Routed_printed_component defines implementation for zero, one, or more Transmission_line_functional_unit_to_routed_printed_component_allocation objects.

4.3.1337 Trimmed_line to Cartesian_point

Each Trimmed_line has end_point defined by exactly one Cartesian_point. Each Cartesian_point defines end_point for zero, one, or more Trimmed_line objects. Each Trimmed_line has start_point defined by exactly one Cartesian_point. Each Cartesian_point defines start_point for zero, one, or more Trimmed_line objects.

4.3.1338 True_profile to Dimension_set

Each True_profile has profile_representation defined by zero or one Dimension_set objects. Each Dimension_set defines profile_representation for zero, one, or more True_profile objects. This assertion is established through a SELECT type.

4.3.1339 True_profile to Ee_document

Each True_profile has profile_representation defined by zero or one Ee_document objects. Each Ee_document defines profile_representation for zero, one, or more True_profile objects. This assertion is established through a SELECT type.

4.3.1340 True_profile to Ee_rule_function

Each True_profile has profile_representation defined by zero or one Ee_rule_function objects. Each Ee_rule_function defines profile_representation for zero, one, or more True_profile objects. This assertion is established through a SELECT type.

4.3.1341 True_profile to Bound_volume_shape

Each True_profile has profile_representation defined by zero or one Bound_volume_shape objects. Each Bound_volume_shape defines profile_representation for zero, one, or more True_profile objects. This assertion is established through a SELECT type.

4.3.1342 True_profile to Physical_feature_or_part_template

Each True_profile has profiled_surface defined by exactly one Physical_feature_or_part_template. Each Physical_feature_or_part_template defines profiled_surface for zero, one, or more True_profile objects.

4.3.1343 Unrouted_conductive_interconnect_element to Layer_connection_point_link

Each Unrouted_conductive_interconnect_element has connected_points defined by one or more Layer_connection_point_link objects. Each Layer_connection_point_link defines connected_points for zero, one, or more Unrouted_conductive_interconnect_element objects.

4.3.1344 Unsupported_passage to Unsupported_passage_template

Each Unsupported_passage has usage_definition defined by exactly one Unsupported_passage_template. Each Unsupported_passage_template defines usage_definition for zero, one, or more Unsupported_passage objects.

4.3.1345 Unsupported_passage_dependent_non_functional_land to Default_unsupported_passage_based_land_physical_template

Each Unsupported_passage_dependent_non_functional_land has usage_definition defined by exactly one Default_unsupported_passage_based_land_physical_template. Each Default_unsupported_passage_based_land_physical_template defines usage_definition for zero, one, or more Unsupported_passage_dependent_non_functional_land objects.

4.3.1346 Unsupported_passage_dependent_non_functional_land to Unsupported_passage

Each Unsupported_passage_dependent_non_functional_land has reference_passage defined by exactly one Unsupported_passage. Each Unsupported_passage defines reference_passage for zero, one, or more Unsupported_passage_dependent_non_functional_land objects.

4.3.1347 Unsupported_passage_template to Default_unsupported_passage_definition

Each Unsupported_passage_template has of_passage_technology defined by exactly one Default_unsupported_passage_definition. Each Default_unsupported_passage_definition defines of_passage_technology for zero, one, or more Unsupported_passage_template objects.

4.3.1348 Usage_concept_usage_relationship to Axis_placement

Each Usage_concept_usage_relationship has associated_usage_placement defined by exactly one Axis_placement. Each Axis_placement defines associated_usage_placement for zero, one, or more Usage_concept_usage_relationship objects.

4.3.1349 Usage_concept_usage_relationship to Shape_definition

Each Usage_concept_usage_relationship has associated_shape_definition defined by exactly one Shape_definition. Each Shape_definition defines associated_shape_definition for zero, one, or more Usage_concept_usage_relationship objects.

4.3.1350 Usage_concept_usage_relationship to Land_template_terminal

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Land_template_terminal objects. Each Land_template_terminal defines associated_usage for zero or one Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1351 Usage_concept_usage_relationship to Component_termination_passage_template_terminal

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Component_termination_passage_template_terminal objects. Each Component_termination_passage_template_terminal defines associated_usage for zero or one Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1352 Usage_concept_usage_relationship to Via_template_terminal

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Via_template_terminal objects. Each Via_template_terminal defines associated_usage for zero or one Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1353 Usage_concept_usage_relationship to Non_feature_shape_element

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Non_feature_shape_element objects. Each Non_feature_shape_element defines associated_usage for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1354 Usage_concept_usage_relationship to Part_feature

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Part_feature objects. Each Part_feature defines associated_usage for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1355 Usage_concept_usage_relationship to Printed_part_template_terminal

Each Usage_concept_usage_relationship has associated_usage defined by zero or one Printed_part_template_terminal objects. Each Printed_part_template_terminal defines associated_usage for one or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1356 Usage_concept_usage_relationship to Part_template_planar_shape

Each Usage_concept_usage_relationship has associating_usage_shape defined by zero or one Part_template_planar_shape objects. Each Part_template_planar_shape defines associating_usage_shape for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1357 Usage_concept_usage_relationship to Physical_unit_planar_shape

Each Usage_concept_usage_relationship has associating_usage_shape defined by zero or one Physical_unit_planar_shape objects. Each Physical_unit_planar_shape defines associating_usage_shape for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1358 Usage_concept_usage_relationship to Part_template_3d_shape

Each Usage_concept_usage_relationship has associating_usage_shape defined by zero or one Part_template_3d_shape objects. Each Part_template_3d_shape defines associating_usage_shape for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1359 Usage_concept_usage_relationship to Part_template_non_planar_2d_shape

Each Usage_concept_usage_relationship has associating_usage_shape defined by zero or one Part_template_non_planar_2d_shape objects. Each Part_template_non_planar_2d_shape defines associating_usage_shape for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1360 Usage_concept_usage_relationship to Physical_unit_3d_shape

Each Usage_concept_usage_relationship has associating_usage_shape defined by zero or one Physical_unit_3d_shape objects. Each Physical_unit_3d_shape defines associating_usage_shape for zero, one, or more Usage_concept_usage_relationship objects. This assertion is established through a SELECT type.

4.3.1361 Usage_concept_usage_relationship to Non_feature_shape_definition

Each Usage_concept_usage_relationship has associating_non_feature_usage_shape defined by exactly one Non_feature_shape_definition. Each Non_feature_shape_definition defines associating_non_feature_usage_shape for zero, one, or more Usage_concept_usage_relationship objects.

4.3.1362 Usage_view_connection_zone_terminal_shape_relationship to Assembly_module_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Assembly_module_terminal objects. Each Assembly_module_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1363 Usage_view_connection_zone_terminal_shape_relationship to Bare_die_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Bare_die_terminal objects. Each Bare_die_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1364 Usage_view_connection_zone_terminal_shape_relationship to Component_termination_passage_template_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Component_termination_passage_template_terminal objects. Each Component_termination_passage_template_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1365 Usage_view_connection_zone_terminal_shape_relationship to Interconnect_module_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Interconnect_module_terminal objects. Each Interconnect_module_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1366 Usage_view_connection_zone_terminal_shape_relationship to Land_template_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Land_template_terminal objects. Each Land_template_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1367 Usage_view_connection_zone_terminal_shape_relationship to Package_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Package_terminal objects. Each Package_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1368 Usage_view_connection_zone_terminal_shape_relationship to Printed_part_template_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Printed_part_template_terminal objects. Each Printed_part_template_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1369 Usage_view_connection_zone_terminal_shape_relationship to Via_template_terminal

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Via_template_terminal objects. Each Via_template_terminal defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1370 Usage_view_connection_zone_terminal_shape_relationship to Part_mounting_feature

Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal defined by zero or one Part_mounting_feature objects. Each Part_mounting_feature defines associating_terminal for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. This assertion is established through a SELECT type.

4.3.1371 Usage_view_connection_zone_terminal_shape_relationship to Axis_placement

Each Usage_view_connection_zone_terminal_shape_relationship has associated_usage_placement defined by exactly one Axis_placement. Each Axis_placement defines associated_usage_placement for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects.

4.3.1372 Usage_view_connection_zone_terminal_shape_relationship to Feature_shape_definition

Each Usage_view_connection_zone_terminal_shape_relationship has associated_connection_zone_shape_definition defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associated_connection_zone_shape_definition for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects. Each Usage_view_connection_zone_terminal_shape_relationship has associating_terminal_shape defined by exactly one Feature_shape_definition. Each Feature_shape_definition defines associating_terminal_shape for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects.

4.3.1373 Usage_view_connection_zone_terminal_shape_relationship to Connection_zone

Each Usage_view_connection_zone_terminal_shape_relationship has associated_usage defined by exactly one Connection_zone. Each Connection_zone defines associated_usage for zero, one, or more Usage_view_connection_zone_terminal_shape_relationship objects.

4.3.1374 Via to Via_template

Each Via has usage_definition defined by exactly one Via_template. Each Via_template defines usage_definition for zero, one, or more Via objects.

4.3.1375 Via_and_contact_size_dependent_land to Via

Each Via_and_contact_size_dependent_land has reference_via defined by exactly one Via. Each Via defines reference_via for zero, one, or more Via_and_contact_size_dependent_land objects.

4.3.1376 Via_and_contact_size_dependent_land to Default_attachment_size_and_via_based_land_physical_template

Each Via_and_contact_size_dependent_land has usage_definition defined by exactly one Default_attachment_size_and_via_based_land_physical_template. Each Default_attachment_size_and_via_based_land_physical_template defines usage_definition for zero, one, or more Via_and_contact_size_dependent_land objects.

4.3.1377 Via_and_contact_size_dependent_non_functional_land to Default_attachment_size_and_via_based_land_physical_template

Each Via_and_contact_size_dependent_non_functional_land has usage_definition defined by exactly one Default_attachment_size_and_via_based_land_physical_template. Each Default_attachment_size_and_via_based_land_physical_template defines usage_definition for zero, one, or more Via_and_contact_size_dependent_non_functional_land objects.

4.3.1378 Via_and_contact_size_dependent_non_functional_land to Via

Each Via_and_contact_size_dependent_non_functional_land has reference_via defined by exactly one Via. Each Via defines reference_via for zero, one, or more Via_and_contact_size_dependent_non_functional_land objects.

4.3.1379 Via_dependent_land to Default_via_based_land_physical_template

Each Via_dependent_land has usage_definition defined by exactly one Default_via_based_land_physical_template. Each Default_via_based_land_physical_template defines usage_definition for zero, one, or more Via_dependent_land objects.

4.3.1380 Via_dependent_land to Via

Each Via_dependent_land has reference_via defined by exactly one Via. Each Via defines reference_via for zero, one, or more Via_dependent_land objects.

4.3.1381 Via_dependent_non_functional_land to Default_via_based_land_physical_template

Each Via_dependent_non_functional_land has usage_definition defined by exactly one Default_via_based_land_physical_template. Each Default_via_based_land_physical_template defines usage_definition for zero, one, or more Via_dependent_non_functional_land objects.

4.3.1382 Via_dependent_non_functional_land to Via

Each Via_dependent_non_functional_land has reference_via defined by exactly one Via. Each Via defines reference_via for zero, one, or more Via_dependent_non_functional_land objects.

4.3.1383 Via_template to Default_via_definition

Each Via_template has of_passage_technology defined by exactly one Default_via_definition. Each Default_via_definition defines of_passage_technology for zero, one, or more Via_template objects.

4.3.1384 Via_template_terminal to Connection_zone

Each Via_template_terminal has terminal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines terminal_connection_zone for zero, one, or more Via_template_terminal objects.

4.3.1385 Via_template_terminal to Inter_stratum_extent

Each Via_template_terminal has disallowed_inter_stratum_extent defined by exactly one Inter_stratum_extent. Each Inter_stratum_extent defines disallowed_inter_stratum_extent for zero, one, or more Via_template_terminal objects.

4.3.1386 Via_template_terminal to Via_template

Each Via_template_terminal has associated_definition defined by exactly one Via_template. Each Via_template defines associated_definition for two or more Via_template_terminal objects.

4.3.1387 Via_terminal to Via

Each Via_terminal has associated_component defined by exactly one Via. Each Via defines associated_component for zero, one, or more Via_terminal objects.

4.3.1388 Via_terminal to Via_template_terminal

Each Via_terminal has definition defined by exactly one Via_template_terminal. Each Via_template_terminal defines definition for zero, one, or more Via_terminal objects.

4.3.1389 Viewing_plane_based_orientation to Viewing_plane_definition

Each Viewing_plane_based_orientation has viewing_plane_specification defined by exactly one Viewing_plane_definition. Each Viewing_plane_definition defines viewing_plane_specification for zero, one, or more Viewing_plane_based_orientation objects.

4.3.1390 Viewing_plane_based_orientation to Tolerance_zone_and_viewing_plane_orienting_relationship

Each Viewing_plane_based_orientation has orienting_relationship defined by exactly one Tolerance_zone_and_viewing_plane_orienting_relationship. Each Tolerance_zone_and_viewing_plane_orienting_relationship defines orienting_relationship for zero, one, or more Viewing_plane_based_orientation objects.

4.3.1391 Viewing_plane_definition to Axis_placement

Each Viewing_plane_definition has viewing_plane_representation defined by exactly one Axis_placement. Each Axis_placement defines viewing_plane_representation for zero, one, or more Viewing_plane_definition objects.

4.3.1392 Viewing_plane_definition to Viewing_plane

Each Viewing_plane_definition has defined_viewing_plane defined by exactly one Viewing_plane. Each Viewing_plane defines defined_viewing_plane for zero, one, or more Viewing_plane_definition objects.

4.3.1393 View_level_non_feature_shape_definition to Part_template_3d_shape

Each View_level_non_feature_shape_definition has model_shape defined by zero or one Part_template_3d_shape objects. Each Part_template_3d_shape defines model_shape for zero, one, or more View_level_non_feature_shape_definition objects. This assertion is established through a SELECT type.

4.3.1394 View_level_non_feature_shape_definition to Physical_unit_3d_shape

Each View_level_non_feature_shape_definition has model_shape defined by zero or one Physical_unit_3d_shape objects. Each Physical_unit_3d_shape defines model_shape for zero, one, or more View_level_non_feature_shape_definition objects. This assertion is established through a SELECT type.

4.3.1395 View_level_non_feature_shape_definition to Part_template_non_planar_2d_shape

Each View_level_non_feature_shape_definition has model_shape defined by zero or one Part_template_non_planar_2d_shape objects. Each Part_template_non_planar_2d_shape defines model_shape for zero, one, or more View_level_non_feature_shape_definition objects. This assertion is established through a SELECT type.

4.3.1396 View_level_non_feature_shape_definition to Part_template_planar_shape

Each View_level_non_feature_shape_definition has model_shape defined by zero or one Part_template_planar_shape objects. Each Part_template_planar_shape defines model_shape for zero, one, or more View_level_non_feature_shape_definition objects. This assertion is established through a SELECT type.

4.3.1397 View_level_non_feature_shape_definition to Physical_unit_planar_shape

Each View_level_non_feature_shape_definition has model_shape defined by zero or one Physical_unit_planar_shape objects. Each Physical_unit_planar_shape defines model_shape for zero, one, or more View_level_non_feature_shape_definition objects. This assertion is established through a SELECT type.

4.3.1398 Volume_shape_intersection to Axis_placement_2d

Each Volume_shape_intersection has planar_shape_origin defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines planar_shape_origin for zero, one, or more Volume_shape_intersection objects.

4.3.1399 Volume_shape_intersection to Axis_placement_3d

Each Volume_shape_intersection has orientation_intersection_plane defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines orientation_intersection_plane for zero, one, or more Volume_shape_intersection objects.

4.3.1400 Volume_shape_intersection to Bound_volume_shape

Each Volume_shape_intersection has projected_volume defined by exactly one Bound_volume_shape. Each Bound_volume_shape defines projected_volume for zero, one, or more Volume_shape_intersection objects.

4.3.1401 Volume_shape_intersection to Planar_shape

Each Volume_shape_intersection has surface_intersection_result defined by exactly one Planar_shape. Each Planar_shape defines surface_intersection_result for zero, one, or more Volume_shape_intersection objects.

4.3.1402 Volume_shape_plane_projection to Axis_placement_2d

Each Volume_shape_plane_projection has planar_shape_origin defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines planar_shape_origin for zero, one, or more Volume_shape_plane_projection objects.

4.3.1403 Volume_shape_plane_projection to Axis_placement_3d

Each Volume_shape_plane_projection has viewing_plane defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines viewing_plane for zero, one, or more Volume_shape_plane_projection objects. Each Volume_shape_plane_projection has orientation_viewing_plane defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines orientation_viewing_plane for zero, one, or more Volume_shape_plane_projection objects.

4.3.1404 Volume_shape_plane_projection to Bound_volume_shape

Each Volume_shape_plane_projection has projected_volume defined by exactly one Bound_volume_shape. Each Bound_volume_shape defines projected_volume for zero, one, or more Volume_shape_plane_projection objects.

4.3.1405 Volume_shape_plane_projection to Planar_shape

Each Volume_shape_plane_projection has surface_projection_result defined by exactly one Planar_shape. Each Planar_shape defines surface_projection_result for zero, one, or more Volume_shape_plane_projection objects.

4.3.1406 Volume_shape_point_projection to Axis_placement_2d

Each Volume_shape_point_projection has planar_shape_origin defined by exactly one Axis_placement_2d. Each Axis_placement_2d defines planar_shape_origin for zero, one, or more Volume_shape_point_projection objects.

4.3.1407 Volume_shape_point_projection to Cartesian_point

Each Volume_shape_point_projection has viewing_point defined by exactly one Cartesian_point. Each Cartesian_point defines viewing_point for zero, one, or more Volume_shape_point_projection objects.

4.3.1408 Volume_shape_point_projection to Axis_placement_3d

Each Volume_shape_point_projection has orientation_viewing_plane defined by exactly one Axis_placement_3d. Each Axis_placement_3d defines orientation_viewing_plane for zero, one, or more Volume_shape_point_projection objects.

4.3.1409 Volume_shape_point_projection to Bound_volume_shape

Each Volume_shape_point_projection has projected_volume defined by exactly one Bound_volume_shape. Each Bound_volume_shape defines projected_volume for zero, one, or more Volume_shape_point_projection objects.

4.3.1410 Volume_shape_point_projection to Planar_shape

Each Volume_shape_point_projection has surface_projection_result defined by exactly one Planar_shape. Each Planar_shape defines surface_projection_result for zero, one, or more Volume_shape_point_projection objects.

4.3.1411 Wire_terminal to Length_data_element

Each Wire_terminal has minimum_wire_terminal_length defined by exactly one Length_data_element. Each Length_data_element defines minimum_wire_terminal_length for zero, one, or more Wire_terminal objects. Each Wire_terminal has maximum_wire_terminal_length defined by exactly one Length_data_element. Each Length_data_element defines maximum_wire_terminal_length for zero, one, or more Wire_terminal objects.

4.3.1412 Wire_terminal to Connection_zone

Each Wire_terminal has internal_connection_zone defined by one or more Connection_zone objects. Each Connection_zone defines internal_connection_zone for zero, one, or more Wire_terminal objects.

4.3.1413 Work_order to Contract

Each Work_order has authorizing_agreement defined by exactly one Contract. Each Contract defines authorizing_agreement for zero, one, or more Work_order objects.

4.3.1414 Work_order to Date_and_time

Each Work_order has release_date defined by zero or one Date_and_time objects. Each Date_and_time defines release_date for zero, one, or more Work_order objects. This assertion is established through a SELECT type. Each Work_order has start_date defined by zero or one Date_and_time objects. Each Date_and_time defines start_date for zero, one, or more Work_order objects. This assertion is established through a SELECT type.

4.3.1415 Work_order to Date

Each Work_order has release_date defined by zero or one Date objects. Each Date defines release_date for zero, one, or more Work_order objects. This assertion is established through a SELECT type. Each Work_order has start_date defined by zero or one Date objects. Each Date defines start_date for zero, one, or more Work_order objects. This assertion is established through a SELECT type.

4.3.1416 Work_order to Person_and_organization

Each Work_order has initiator defined by zero or one Person_and_organization objects. Each Person_and_organization defines initiator for zero, one, or more Work_order objects. This assertion is established through a SELECT type.

4.3.1417 Work_order to Organization

Each Work_order has initiator defined by zero or one Organization objects. Each Organization defines initiator for zero, one, or more Work_order objects. This assertion is established through a SELECT type.

4.3.1418 Work_order to Ee_approval

Each Work_order has work_order_approval defined by exactly one Ee_approval. Each Ee_approval defines work_order_approval for zero, one, or more Work_order objects.

4.3.1419 Work_order to Ee_text

Each Work_order has additional_data defined by exactly one Ee_text. Each Ee_text defines additional_data for zero, one, or more Work_order objects. Each Work_order has analysis_data defined by exactly one Ee_text. Each Ee_text defines analysis_data for zero, one, or more Work_order objects.

4.3.1420 Work_order to Ee_product_version

Each Work_order has resulting_product defined by one or more Ee_product_version objects. Each Ee_product_version defines resulting_product for zero, one, or more Work_order objects.

4.3.1421 Work_order to Work_request

Each Work_order has referenced_work_request defined by one or more Work_request objects. Each Work_request defines referenced_work_request for zero or one Work_order objects.

4.3.1422 Work_request to Person_and_organization

Each Work_request has initiator defined by zero or one Person_and_organization objects. Each Person_and_organization defines initiator for zero, one, or more Work_request objects. This assertion is established through a SELECT type. Each Work_request has request_recipient defined by zero or one Person_and_organization objects. Each Person_and_organization defines request_recipient for zero, one, or more Work_request objects. This assertion is established through a SELECT type.

4.3.1423 Work_request to Organization

Each Work_request has initiator defined by zero or one Organization objects. Each Organization defines initiator for zero, one, or more Work_request objects. This assertion is established through a SELECT type. Each Work_request has request_recipient defined by zero or one Organization objects. Each Organization defines request_recipient for zero, one, or more Work_request objects. This assertion is established through a SELECT type.

4.3.1424 Work_request to Evaluated_characteristic

Each Work_request has reason defined by zero or one Evaluated_characteristic objects. Each Evaluated_characteristic defines reason for zero, one, or more Work_request objects. This assertion is established through a SELECT type.

4.3.1425 Work_request to Ee_text

Each Work_request has reason defined by zero or one Ee_text objects. Each Ee_text defines reason for zero, one, or more Work_request objects. This assertion is established through a SELECT type. Each Work_request has description defined by exactly one Ee_text. Each Ee_text defines description for zero, one, or more Work_request objects.

4.3.1426 Work_request to Date_and_time

Each Work_request has request_date defined by zero or one Date_and_time objects. Each Date_and_time defines request_date for zero, one, or more Work_request objects. This assertion is established through a SELECT type.

4.3.1427 Work_request to Date

Each Work_request has request_date defined by zero or one Date objects. Each Date defines request_date for zero, one, or more Work_request objects. This assertion is established through a SELECT type.

4.3.1428 Work_request to Ee_product_version

Each Work_request has referenced_product defined by one or more Ee_product_version objects. Each Ee_product_version defines referenced_product for zero, one, or more Work_request objects.

4.3.1429 Work_request to Ee_approval

Each Work_request has work_request_approval defined by exactly one Ee_approval. Each Ee_approval defines work_request_approval for zero, one, or more Work_request objects.

5 Application interpreted model

5.1 Mapping table

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). The mapping table is organized in five columns.

Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name> . <attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

Column 3) Source: For those AIM elements that are interpreted from the integrated resources or the application interpreted constructs, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part. Entities or types that are defined within the integrated resources have an AIC as the source reference if the use of the entity or type for the mapping is within the scope of the AIC.

Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. Global rules are given for each application object or AIM element to which they apply. The number in the rules column refers to the subclause number n of 5.2.4.n where the global rule is defined.

Column 5) Reference path: To describe fully the mapping of an application object, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths the following notational conventions apply:

- a) [] : enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;
- b) () : enclosed section constrains multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

- c) `{ }` : enclosed section constrains the reference path to satisfy an information requirement;
- d) `<>` : enclosed section constrains at one or more required reference path;
- e) `| |` : enclosed section constrains the supertype entity;
- f) `->` : attribute references the entity or select type given in the following row;
- g) `<-` : entity or select type is referenced by the attribute in the following row;
- h) `[ i ]` : attribute is an aggregation of which a single member is given in the following row;
- i) `[ n ]` : attribute is an aggregation of which member n is given in the following row;
- j) `=>` : entity is a supertype of the entity given in the following row;
- k) `<=` : entity is a subtype of the entity given in the following row;
- l) `=` : the string, select, or enumeration type is constrained to a choice or value;
- m) `\` : the reference path expression continues on the next line.

Table 1 – Mapping table for advanced_boundary_representation UoF

Application element	AIM element	Source	Rules	Reference path
ADVANCED_B_REP	advanced_brep_shape_- representation	514		

Table 2 – Mapping table for altered_package UoF

Application element	AIM element	Source	Rules	Reference path
ALTERED_PACKAGE	(package) (externally_defined_package) (library_defined_package)	210 210 210		(package <= physical_unit <= product_definition) (externally_defined_package <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.description = 'altered package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}]}
altered_package to package (as package_to_be_altered)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <=

Table 2 – Mapping table for altered_package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'package alteration'} product_definition_relationship product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
altered_package to external_definition (as of_geometric_status)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition

Table 2 – Mapping table for altered_package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'externally defined geometric status'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition
altered_package to length_data_element (as modified_terminal_separation)	PATH			(package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition

Table 2 – Mapping table for altered_package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'modified terminal separation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> </pre>
altered_package to predefined_geometric_status (as of_geometric_status)	PATH			

Table 2 – Mapping table for altered_package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation representation.name = 'predefined geometric status'} representation.items[i] -> representation_item {(representation_item.name = 'standard datum condition') (representation_item.name = 'undefined') (representation_item.name = 'perpendicular primary datum axis') (representation_item.name = 'perpendicular primary datum plane')} representation_item => descriptive_representation_item </pre>
ALTERED_PACKAGE_TERMINAL	altered_package_terminal	210		<pre> altered_package_terminal <= package_terminal <= shape_aspect </pre>
altered_package_terminal to package_terminal (as terminal_to_be_altered)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal to be altered'} shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => package_terminal </pre>
altered_package_terminal to altered_package (as associated_definition)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 2 – Mapping table for altered_package UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.description = 'altered package']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)</pre>

Table 3 – Mapping table for altered_part UoF

Application element	AIM element	Source	Rules	Reference path
ALTERED_PACKAGE	(package) (externally_defined_package) (library_defined_package)	210 210 210		(package <= physical_unit <= product_definition) (externally_defined_package <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.description = 'altered package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}]}
altered_package to package (as package_to_be_altered)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <=

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'package alteration'} product_definition_relationship product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
altered_package to external_definition (as of_geometric_status)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'externally defined geometric status'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition
altered_package to length_data_element (as modified_terminal_separation)	PATH			(package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'modified terminal separation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> </pre>
altered_package to predefined_geometric_status (as of_geometric_status)	PATH			<pre> (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> </pre>

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation representation.name = 'predefined geometric status'} representation.items[i] -> representation_item {(representation_item.name = 'standard datum condition') (representation_item.name = 'undefined') (representation_item.name = 'perpendicular primary datum axis') (representation_item.name = 'perpendicular primary datum plane')}} representation_item => descriptive_representation_item </pre>
ALTERED_PACKAGE_TERMINAL	altered_package_terminal	210		<pre> altered_package_terminal <= package_terminal <= shape_aspect </pre>
altered_package_terminal to package_terminal (as terminal_to_be_altered)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal to be altered'} shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => package_terminal </pre>
altered_package_terminal to altered_package (as associated_definition)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition {product_definition [product_definition.description = 'altered package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
ALTERED_PACKAGED_PART	(packaged_part) (externally_defined_packaged_ part) (library_defined_packaged_part)	210 210 210		(packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_part <= library_defined_physical_unit <=

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.description = 'altered packaged part'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} (packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'base packaged part'} product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'}
altered_packaged_part to packaged_part (as base_packaged_part)	PATH			(packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'base packaged part'} product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'}

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (packaged_part) (externally_defined_physical_unit => externally_defined_packaged_part) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part) </pre>
altered_packaged_part to altered_package (as used_package)	PATH			<pre> (packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition <- product_definition_relationship.related_ - product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'used package'} product_definition_relationship.related_ - product_definition -> {product_definition [product_definition.description = 'altered package']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'']] product_definition => physical_unit => </pre>

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
ENGINEERING_MAKE_FROM	design_make_from_relationship	210		design_make_from_relationship <= product_definition_relationship
engineering_make_from to ee_product_definition (as base_design)	PATH		338	design_make_from_relationship <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition => (product_definition_with_associated_documents) (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => externally_defined_physical_unit) library_defined_physical_unit) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => externally_defined_functional_unit) library_defined_functional_unit)
engineering_make_from to ee_product_definition (as resultant_design)	PATH			design_make_from_relationship <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => (product_definition_with_associated_documents) (physical_unit)

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)</pre>
PART_FEATURE_MAKE_- FROM_RELATIONSHIP	make_from_part_feature_- relationship	210		<pre>{[make_from_part_feature_relationship <= shape_aspect_relationship] [make_from_part_feature_relationship <= shape_aspect]}</pre>
part_feature_make_from_- relationship to engineering_make_from (as associated_make_from)	PATH			<pre>make_from_part_feature_relationship <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => design_make_from_relationship</pre>
part_feature_make_from_- relationship to part_feature (as reusable_feature)	PATH			<pre>make_from_part_feature_relationship <= shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect</pre>

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.description = 'part group feature'] (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect =>

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
part_feature_make_from_- relationship to part_feature (as resultant_feature)	PATH			make_from_part_feature_relationship <= shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'] (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface')

Table 3 – Mapping table for altered_part UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}} </pre>

Table 3 – Mapping table for altered_part UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path

Table 4 – Mapping table for analytical_model UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_MODEL	analytical_model	210		analytical_model <= representation
name	PATH			analytical_model <= representation representation.name
analytical_model to ee_document (as reference_document)	PATH			analytical_model <= representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'reference document'}
analytical_model to analytical_model_definition (as associated_definition)	PATH			analytical_model <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition ->

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>property_definition</code> <code>property_definition.definition -></code> <code>characterized_definition</code> <code>characterized_definition = characterized_-</code> <code>product_definition</code> <code>characterized_product_definition</code> <code>characterized_product_definition = product_-</code> <code>definition</code> <code>product_definition =></code> <code>analytical_model_definition</code>
analytical_model to language_reference_manual (as representation_language)	PATH			<code>analytical_model <=</code> <code>representation</code> <code>document_assigned_item = representation</code> <code>document_assigned_item <-</code> <code>applied_document_reference.items[i]</code> <code>applied_document_reference <=</code> <code>document_reference</code> <code>document_reference.assigned_document -></code> <code>{document</code> <code>document.kind -></code> <code>document_type</code> <code>document_type.product_data_type = 'language</code> <code>reference_manual'}</code> <code>document =></code> <code>ee_specification</code>
analytical_model to ee_specification (as source_code)	PATH			<code>analytical_model <=</code> <code>representation</code> <code>document_assigned_item = representation</code> <code>document_assigned_item <-</code> <code>applied_document_reference.items[i]</code> <code>applied_document_reference <=</code> <code>document_reference</code> <code>document_reference.assigned_document -></code> <code>{document</code>

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document.kind -> document_type document_type.product_data_type = 'source code'} document => ee_specification
analytical_model to model_parameter (as parameter)	PATH			analytical_model <= representation representation.items[i] -> representation_item => model_parameter
ANALYTICAL_MODEL_DEFINITION	analytical_model_definition	210		analytical_model_definition <= product_definition {product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <= product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'simulation model'}
ANALYTICAL_MODEL_MAKE_FROM_RELATIONSHIP	analytical_model_make_from_ - relationship	210		{[analytical_model_make_from_relationship <= representation_relationship] [analytical_model_make_from_relationship <= representation]}
analytical_model_make_from_ - relationship to analytical_model (as reusable_model)	PATH			analytical_model_make_from_relationship <= representation_relationship representation_relationship.rep_1 -> representation => analytical_model

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
analytical_model_make_from_- relationship to analytical_model (as resultant_model)	PATH			analytical_model_make_from_relationship <= representation_relationship representation_relationship.rep_2 -> representation => analytical_model
ANALYTICAL_MODEL_PORT	analytical_model_port	210	342, 5	analytical_model_port <= representation {representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition (characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect) (characterized_definition = characterized_ object characterized_object))}
distributed_properties	descriptive_representation_item	45	342	analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'distributed property'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
nominal_signal_flow_direction	descriptive_representation_item	45		<pre> analytical_model_port <= representation {representation.items[i] -> representation_item representation_item.name = 'nominal signal flow direction'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'input direction') (descriptive_representation_item.description = 'output direction') (descriptive_representation_item.description = 'bidirectional') (descriptive_representation_item.description = 'unknown direction') (descriptive_representation_item.description = 'not applicable')}} </pre>
port_name	representation.name	43	342	<pre> analytical_model_port <= representation representation.name </pre>
port_type	descriptive_representation_item	45	342	<pre> analytical_model_port <= representation {representation.items[i] -> representation_item representation_item.name = 'port type'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'string property type') (descriptive_representation_item.description = 'logical property type') (descriptive_representation_item.description = 'numeric property type')} </pre>

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'physical property type') (descriptive_representation_item.description = 'boolean property type')}
unit_of_measure	(named_unit) (derived_unit)	41 41	342	analytical_model_port <= representation representation.context_of_items -> representation_context => global_unit_assigned_context global_unit_assigned_context.units[i] -> unit (unit = named_unit named_unit) (unit = derived_unit derived_unit)
analytical_model_port to analytical_model (as accessed_analytical_model)	PATH		342	analytical_model_port <= representation <= representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'access mechanism'} representation_relationship.rep_1 -> representation => analytical_model
ANALYTICAL_MODEL_- PORT_MAKE_FROM_RELA- TIONSHIP	make_from_model_port_- relationship	210		{[make_from_model_port_relationship <= representation_relationship] [make_from_model_port_relationship <= representation]}
analytical_model_port_make_- from_relationship to analytical_model_make_from_- relationship (as associated_make_from)	PATH			make_from_model_port_relationship <= representation <= representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'associated make from'} representation_relationship.rep_1 -> representation => analytical_model_make_from_relationship

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
analytical_model_port_make_- from_relationship to analytical_model_port (as reusable_feature)	PATH			make_from_model_port_relationship <= representation_relationship representation_relationship.rep_1 -> representation => analytical_model_port
analytical_model_port_make_- from_relationship to analytical_model_port (as resultant_feature)	PATH			make_from_model_port_relationship <= representation_relationship representation_relationship.rep_2 -> representation => analytical_model_port
ANALYTICAL_MODEL_- SCALAR_PORT	analytical_model_port	210		analytical_model_port <= representation {representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group (group.name = 'scalar port') (group.name = 'digital scalar port')}
ANALYTICAL_MODEL_VEC- TOR_PORT	analytical_model_port	210	6, 342	analytical_model_port <= representation {representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group (group.name = 'vector port') (group.name = 'digital vector port')}
size	measure_representation_item	45	342, 109, 105	analytical_model_port <= representation representation.items[i] ->

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'size'} representation_item => measure_representation_item measure_representation_item <= measure_with_unit {measure_with_unit.unit_component -> unit unit = named_unit named_unit => context_dependent_unit} measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure </pre>
DIGITAL_ANALYTICAL_- MODEL_PORT	analytical_model_port	210	342	<pre> analytical_model_port <= representation representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group.name = 'digital analytical model port'} analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'distributed property'} representation_item => descriptive_representation_item </pre>
nominal_signal_flow_direction	descriptive_representation_item	45	342, 109	

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{descriptive_representation_item (descriptive_representation_item.description = 'input direction') (descriptive_representation_item.description = 'output direction') (descriptive_representation_item.description = 'bidirectional') (descriptive_representation_item.description = 'unknown direction')}}
LANGUAGE_REFERENCE_- MANUAL	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'language reference manual'}
MODEL_PARAMETER	(model_parameter) ([model_parameter] [measure_representation_item]) ([model_parameter] [descriptive_representation_item])	210 210 41 210 43	109, 211	(model_parameter <= representation_item) ([model_parameter <= representation_item] [measure_representation_item <= representation_item]) ([model_parameter <= representation_item] [descriptive_representation_item <= representation_item]) {representation_item group_assigned_item = representation_item group_assigned_item <= applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => parameter_type}

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
type_name	representation_item.name	43	109	model_parameter <= representation_item representation_item.name
unit_of_measure	measure_with_unit.unit_ component	41		model_parameter <= representation_item => measure_representation_item <= measure_with_unit measure_with_unit.unit_component
model_parameter to characteristic (as default_value)	PATH			model_parameter <= representation_item <= representation.items[i] representation <= representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_2 -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <= applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type} model_parameter <= representation_item <= representation.items[i] representation <=
model_parameter to coordinated_characteristic (as default_value)	PATH			

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_1 -> representation => coordinated_representation_item </pre>
model_parameter to coordinated_characteristic (as valid_range)	PATH			<pre> model_parameter <= representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'valid range'} representation_relationship.rep_1 -> representation representation.items[i] -> representation_item => coordinated_representation_item </pre>
model_parameter to ee_document (as reference_document)	PATH			<pre> model_parameter <= representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> (document) (document => ee_specification) </pre>

Table 4 – Mapping table for analytical_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
TERMINAL_PAIR_PORT_- EQUIVALENCE_RELATION- SHIP	representation_relationship	43	359	{representation_relationship representation_relationship.name = 'terminal pair port equivalence relationship'}
terminal_pair_port_equivalence_- relationship to analytical_model_port (as equivalent_port)	PATH		342	representation_relationship representation_relationship.rep_2 -> {representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition (characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect) (characterized_definition = characterized_- object characterized_object)} representation => analytical_model_port
terminal_pair_port_equivalence_- relationship to analytical_model_scalar_port (as scalar_ports)	PATH		342	representation_relationship representation_relationship.rep_1 -> {representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <=

Table 4 – Mapping table for analytical_model UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>group_assignment group_assignment.assigned_group -> group (group.name = 'scalar port') (group.name = 'digital scalar port') [representation representation.name = 'analytical model scalar port'] [representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'scalar port list'} representation_relationship.rep_2 -> representation representation.name = 'analytical model scalar port'}} representation => analytical_model_port</pre>

Table 5 – Mapping table for analytical_representation UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_REPRESENTATION	analytical_representation	210		analytical_representation <= representation
analytical_representation_name	representation.name	43		analytical_representation <= representation representation.name
analytical_representation to analytical_model (as model_representation)	PATH			analytical_representation <= representation <- representation_relationship representation_relationship representation_relationship.rep_1 -> representation => analytical_model
analytical_representation to parameter_assignment (as model_parameter_assignment)	PATH			analytical_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => parameter_assignment

Table 6 – Mapping table for assembly_bill_of_material UoF

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_PRODUCT	alternate_product_relationship	44		{alternate_product_relationship alternate_product_relationship.basis = 'alternate_product'}
alternate_product to ee_product (as defined_alternate_product)	PATH			alternate_product_relationship alternate_product_relationship.alternate -> product
ALTERNATE_SELECT_PROD- UCT	alternate_product_relationship	44		{alternate_product_relationship (alternate_product_relationship.basis = 'alternate_product') (alternate_product_relationship.basis = 'test select product') (alternate_product_relationship.basis = 'assembly_alternate_product')}
alternate_select_product to contract (as authorizing_agreement)	PATH			alternate_product_relationship contract_assigned_item = alternate_product_- relationship contract_assigned_item <= applied_contract_assignment.items[i] applied_contract_assignment <= contract_assignment contract_assignment.assigned_contract -> contract
alternate_select_product to ee_product (as defined_alternate_product)	PATH			alternate_product_relationship alternate_product_relationship.alternate -> product
alternate_select_product to ee_product (as defining_base_product)	PATH			alternate_product_relationship alternate_product_relationship.base -> product

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_ALTERNATE_- PRODUCT	alternate_product_relationship	44		{alternate_product_relationship alternate_product_relationship.basis = 'assembly alternate product'}
assembly_alternate_product to physical_unit_design_view (as basis_context)	PATH		14, 268, 338, 339, 354	alternate_product_relationship alternate_product_relationship.base -> product <- product_definition_formation.of_product product_definition_formation <- product_definition_formation product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
ASSEMBLY_COMPOSITION_- RELATIONSHIP	assembly_component_usage	44	354, 339, 15	{assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly composition'}
assembly_composition_- relationship to assembly_component (as component)	PATH		354, 339, 338, 340, 333	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{product_definition => component_definition}
assembly_composition_- relationship to ee_material (as assembly_bonding_agent)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= material_designation.definitions[i] material_designation
assembly_composition_- relationship to physical_unit_design_view (as assembly)	PATH		354, 339, 338	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
assembly_composition_- relationship to security_classification (as security_code)	PATH			assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <= applied_security_classification_- assignment.items[i] applied_security_classification_assignment <=

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification
ASSEMBLY_MATERIAL_COM- POSITION_RELATIONSHIP	(next_assembly_usage_ occurrence) ([next_assembly_usage_ occurrence] [quantified_assembly_ component_usage])	44 44 44		(next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}) ([quantified_assembly_component_usage] {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}})]
as_required	PATH	41		quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit measure_with_unit.value_component -> measure_value measure_value = descriptive_measure {(descriptive_measure = 'as required') (descriptive_measure = 'quantity specified'))}
assembly_material_composition_ relationship to data_element (as quantity)	PATH			quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_material_composition_- relationship to material_definition (as component_material)	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'material definition'} </pre>
assembly_material_composition_- relationship to physical_unit_design_view (as assembly)	PATH		338	<pre> next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition product_definition => physical_unit </pre>
assembly_material_composition_- relationship to security_classification (as security_code)	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
NEXT_HIGHER_ASSEMBLY_- RELATIONSHIP	<pre> (next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage]) </pre>	44 44 44	213, 214	<pre> (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=]) assembly_component_usage </pre>

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
reference_designation	PATH			next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator
next_higher_assembly_- relationship to data_element (as quantity)	PATH			next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage -> measure_with_unit
PHYSICAL_COMPONENT	component_definition	210		component_definition <= product_definition {[product_definition product_definition.description != 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']}]}
physical_component to physical_unit (as usage_definition)	PATH			component_definition <= product_definition <= product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design')}

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(application_context_element.name = 'physical design usage') product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit)</pre>
PROMISSORY_USAGE_IN_PRODUCT_MODEL_RELATIONSHIP	applied_promissory_usage_in_product_model_assignment	210		<pre>{applied_promissory_usage_in_product_model_assignment =<= group_assignment group_assignment.assigned_group -> group => promissory_usage_in_product_model}</pre>
promissory_usage_in_product_model_relationship to Ee_product_definition (as component)	PATH			<pre>applied_promissory_usage_in_product_model_assignment =<= applied_promissory_usage_in_product_model_assignment.items[i] -> promissory_usage_in_product_model_assigned_item promissory_usage_in_product_model_assigned_item = product_definition (product_definition => product_definition_with_associated_documents) (product_definition) (product_definition => evaluation_product_definition) (product_definition => physical_unit) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition =></pre>

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				analytical_model_definition) (product_definition => footprint_definition) (product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => stratum) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit)
promissory_usage_in_product_model_relationship to ee_product_model (as product_model)	PATH			applied_promissory_usage_in_product_model_assignment applied_promissory_usage_in_product_model_assignment.items[i] -> promissory_usage_in_product_model_assigned_item promissory_usage_in_product_model_assigned_item = product_concept
promissory_usage_in_product_model_relationship to security_classification (as security_code)	PATH			applied_promissory_usage_in_product_model_assignment security_classification_assigned_item = applied_promissory_usage_in_product_model_assignment security_classification_assigned_item <- applied_security_classification_assignment.items[i] applied_security_classification_assignment <= security_classification_assignment

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment.assigned_ security_classification -> security_classification
PROMISSORY_USAGE_RELATIONSHIP	promissory_usage_occurrence	44	275, 276	
promissory_usage_relationship to physical_unit (as component)	PATH		354, 339, 338	promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) library_defined_physical_unit
promissory_usage_relationship to physical_unit_design_view (as assembly)	PATH			promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition ->

Table 6 – Mapping table for assembly_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit </pre>
promissory_usage_relationship to security_classification (as security_code)	PATH			<pre> promissory_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
SUBSTITUTE_PRODUCT	product	41	331	
substitute_product to assembly_- composition_relationship (as substitutable_component)	PATH		268, 339, 354	<pre> product <- product_definition_formation.of_product product_definition_formation <- product_definition_formation product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'assembly_composition'} product_definition_relationship => product_definition_usage => assembly_component_usage </pre>

Table 6 – Mapping table for assembly_bill_of_material UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
substitute_product to ee_product (as defined_altermate_product)	PATH			<pre> {assembly_component_usage <- assembly_component_usage_substitute.base} product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship => product_definition_usage => assembly_component_usage <- substitute.substitute} product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition product_definition.formation -> product_definition.formation product_definition_formation.of_product -> product </pre>
TEST_SELECT_PRODUCT	altermate_product_relationship	44		<pre> {altermate_product_relationship altermate_product_relationship.basis = 'test select product'} </pre>

Table 7 – Mapping table for assembly_component UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_COMPONENT	(component_definition)	210	333, 340, 338	<pre>(component_definition <= (product_definition {[product_definition product_definition.description != 'printed component']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'])) (product_definition {[product_definition product_definition.description = 'printed component']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence'])))</pre>
assembly_component to footprint_definition (as usage_definition)	PATH			<pre>component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'usage definition'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition</pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition product_definition => footprint_definition
assembly_component to padstack_definition (as usage_definition)	PATH			component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'usage definition'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => padstack_definition
assembly_component to physical_unit (as usage_definition)	PATH		338	component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.relatng_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design')}

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(application_context_element.name = 'physical design usage') product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit)</pre>
ASSEMBLY_MODULE_COMPONENT	component_definition	210	338, 268	<pre>component_definition <= product_definition {product_definition [product_definition product_definition.description != 'laminated component'] [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']} }</pre>
assembly_module_component to assembly_module_usage_view (as usage_definition)	PATH		338, 268	<pre>component_definition <= product_definition <- product_definition.relationship.related_ - product_definition {product_definition.relationship product_definition.relationship.name = 'instantiated part'} product_definition.relationship</pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']}] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
ASSEMBLY_MODULE_- MACRO_COMPONENT	component_definition	210		<pre> component_definition <= product_definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']}] </pre>
assembly_module_macro_- component to assembly_module (as design_definition)	PATH			<pre> component_definition <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'design definition'} </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)
ASSEMBLY_MODULE_- MACRO_COMPONENT_JOIN_- TERMINAL	assembly_module_macro_- component_join_terminal	210		assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'assembly module component terminal'] [shape_aspect.product_definitional = .TRUE.]}
assembly_module_macro_- component_join_terminal to assembly_module_macro_- component (as associated_component)	PATH			assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']} product_definition => component_definition
assembly_module_macro_- component_join_terminal to assembly_module_join_terminal (as definition)	PATH			assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relying_shape_ aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'assembly module' shape_aspect => assembly_module_join_terminal => assembly_module_join_terminal
BARE_DIE_COMPONENT	component_definition	210		component_definition <= product_definition {product_definition [product_definition.description = 'bare die component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']}
bare_die_component to bare_die (as usage_definition)	PATH		338	component_definition <= product_definition <- product_definition_relationship.related_- product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'}

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (bare_die) (externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die) </pre>
CABLE_COMPONENT	cable_component	210		<pre> cable_component <= component_definition <= product_definition {product_definition product_definition.description != 'laminated' component'} </pre>
INTERCONNECT_MODULE_- COMPONENT	component_definition	210	338, 268	<pre> component_definition <= product_definition {product_definition [product_definition product_definition.description != 'laminated' component']} [product_definition.name = 'interconnect' module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence'}} </pre>
interconnect_module_component to interconnect_module_usage_view (as usage_definition)	PATH		338, 268	<pre> component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.relatiing_ product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
INTERCONNECT_MODULE_- MACRO_COMPONENT	component_definition	210		<pre> component_definition <= product_definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'macro occurrence']} </pre>
interconnect_module_macro_- component to interconnect_module (as design_definition)	PATH			<pre> component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'design definition'} </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
INTERFACE_COMPONENT	interface_component_definition	210	338, 333, 340	<pre> interface_component_definition <= component_definition <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'} {product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'}} </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship product_definition_relationship.relatiing_- product_definition -> [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'] [product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)]] </pre>
PACKAGED_COMPONENT	packaged_component	210		<pre> packaged_component <= component_definition <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'} </pre>
packaged_component to packaged_part (as usage_definition)	PATH		338	<pre> packaged_component <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.relatiing_- product_definition -> </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (packaged_part) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part) (externally_defined_physical_unit => externally_defined_packaged_part) </pre>
packaged_component to altered_package (as selected_package_altername)	PATH			<pre> packaged_component <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'package_altername'} product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition [product_definition.description = 'altered package']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) (externally_defined_physical_unit => externally_defined_package) </pre>
PACKAGED_CONNECTOR_- COMPONENT	packaged_component	210	338	<pre> packaged_component <= component_definition <= product_definition {[product_definition.description = 'packaged connector component']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']} </pre>
packaged_connector_component to packaged_connector (as usage_definition)	PATH		338	<pre> packaged_component <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship.related_- product_definition -> product_definition => physical_unit => (packaged_part => packaged_connector) (externally_defined_physical_unit => </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_packaged_part => externally_defined_packaged_connector) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector)
PHYSICAL_COMPONENT	component_definition	210		component_definition <= product_definition {[product_definition product_definition.description != 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']}]
physical_component to physical_unit (as usage_definition)	PATH			component_definition <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')}]

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit) </pre>
ROUTED_INTERCONNECT_- COMPONENT	component_definition	210	333, 340, 338	<pre> component_definition <= product_definition {[product_definition product_definition.description = 'routed interconnect component'] [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']} </pre>
routed_interconnect_component to curve (as routed_centrelines_shape)	PATH			<pre> component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect shape_definition = shape_aspect shape_definition </pre>

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation.items[i] -> representation_item => geometric_representation_item => curve
SHIELD	(discrete_shield)	210	338, 333	(discrete_shield <= component_definition <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'})
shield to assembly_component (as shielded_item)	PATH		338, 340, 333	(discrete_shield <= component_definition <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'shielded_item'}) product_definition_relationship product_definition_relationship.relatiing_ - product_definition -> (product_definition => component_definition))

Table 7 – Mapping table for assembly_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shield to component_feature (as shielded_item)	PATH		338, 340, 333	<pre>(discrete_shield <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'shielded item'}) product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape (shape_aspect => (component_terminal) (component_interface_terminal) (laminate_component_interface_terminal) (component_feature)))</pre>
shield to physical_connectivity_definition (as shielded_item)	PATH		338, 340, 333	<pre>(discrete_shield <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship</pre>

Table 7 – Mapping table for assembly_component UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.name = 'shielded item'} product_definition_relationship product_definition_relationship.relatiing_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => physical_connectivity_definition) </pre>
THERMAL_COMPONENT	(thermal_component)	210	333, 340, 338	<pre> (thermal_component <= component_definition <= product_definition) </pre>

Table 8 – Mapping table for assembly_component_grouping UoF

Application element	AIM element	Source	Rules	Reference path
ARRAY_PLACEMENT_GROUP	(array_placement_group_- component_definition)	210	338, 333	(array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_ product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']})
ASSEMBLY_COMPONENT_ PLACEMENT_LINK	(assembly_group_component_ definition_placement_link)	210		([{assembly_group_component_definition_ placement_link <= product_definition] [assembly_group_component_definition_ placement_link <= product_definition_relationship]})

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_component_- placement_link to assembly_component (as precedent_element)	PATH			(assembly_group_component_definition_- placement_link <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition => component_definition)
assembly_component_- placement_link to assembly_component (as subsequent_element)	PATH			(assembly_group_component_definition_- placement_link <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition)
ASSEMBLY_GROUP_COMPO- NENT	(assembly_group_component_- definition)	210	338, 333	(assembly_group_component_definition <= component_definition <= product_definition {product_definition <- product_definition_relationship.relatiing_- product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_- product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']})

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERFACED_GROUP_COMPONENT	interfaced_group_component_definition	210		interfaced_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition
interfaced_group_component to interface_component (as group_interface_component)	PATH		338, 333	interfaced_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <- product_definition.relationship.related_product_definition {product_definition_relationship {product_definition_relationship.name = 'group interface component'}} product_definition_relationship.related_product_definition -> product_definition {product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element {application_context_element.name = 'physical occurrence'}}} product_definition => component_definition => interface_component_definition
LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT	(linear_array_placement_group_component_definition)	210		(linear_array_placement_group_component_definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition <- product_definition_relationship.relatiing_- product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_- product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']}) </pre>
linear_array_placement_group_- component to assembly_component_- placement_link (as element)	PATH			<pre> (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <- product_definition_relationship.relatiing_- product_definition {product_definition_relationship product_definition_relationship.name = 'element'} product_definition_relationship.related_- product_definition -> product_definition => assembly_group_component_definition_- placement_link) </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> parametric_representation_context[] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric_context']] representation.items[i] -> representation_item => {representation_item representation_item.name = 'initial element positive x displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value.component </pre>
linear_array_placement_group_- component to length_data_element (as location_uncertainty)	PATH			<pre> (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) property_definition.definition property_definition <- {property_definition property_definition.name = 'linear array placement_group_properties'} </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation.used_ representation -> representation {representation representation.name = 'linear array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_- context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit </pre>
linear_array_placement_group_- component to length_data_element (as terminus_node_positive_x_- displacement)	PATH			<pre> (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <-> property_definition.definition property_definition <- {property_definition property_definition.name = 'linear array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'linear array properties'] [representation representation.context_of_items -> representation_context [representation_context => representation_context => global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']} representation.items[i] -> representation_item =>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'terminus element positive x displacement'} measure_representation_item <= {measure_with_unit length_measure_with_unit} measure_with_unit.value_component </pre>
LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_LINK	(linear_array_component_definition_link)	210		<pre> ({linear_array_component_definition_link <= product_definition} [linear_array_component_definition_link <= product_definition.relationship]) </pre>
linear_array_placement_group_component_link to linear_array_placement_group_component (as precedent_element)	PATH			<pre> (linear_array_component_definition_link <= product_definition.relationship product_definition.relationship.related_ product_definition -> product_definition => component_definition => assembly_group_component_definition => array_placement_group_component_definition => linear_array_placement_group_component_ definition) </pre>
linear_array_placement_group_component_link to linear_array_placement_group_component (as subsequent_element)	PATH			<pre> (linear_array_component_definition_link <= product_definition.relationship product_definition.relationship.related_ product_definition -> product_definition => component_definition => assembly_group_component_definition => array_placement_group_component_definition => linear_array_placement_group_component_ definition) </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_COMPONENT	component_definition	210		<pre> component_definition <= product_definition {[product_definition product_definition.description != 'laminated component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']] </pre>
physical_component to physical_unit (as usage_definition)	PATH			<pre> component_definition <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.related_ - product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit) </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
RECTANGULAR_ARRAY_PLACEMENT_GROUP_COMPONENT	(rectangular_array_placement_group_component_definition)	210		<pre> (rectangular_array_placement_group_component_definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <= {product_definition <- product_definition_relationship.relationship_ product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.relationship.related_ product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'}}) </pre>
rectangular_array_placement_group_component to linear_array_placement_group_component link (as element)	PATH			<pre> (rectangular_array_placement_group_component_definition <= definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <- product_definition_relationship.relationship_ product_definition {product_definition_relationship </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.name = 'element'} product_definition_relationship.related_ product_definition -> product_definition => linear_array_component_definition_link) </pre>
rectangular_array_placement_ group_component to length_data_element (as initial_row_positive_y_ displacement)	PATH			<pre> (rectangular_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <-) property_definition.definition property_definition <- {property_definition property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context representation_context => </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric_context']] representation.items[i] -> representation_item => {representation_item representation_item.name = 'initial row positive y displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component (rectangular_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition </pre>
rectangular_array_placement_ group_component to length_data_element (as location_uncertainty)	PATH			

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- {property_definition property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation {representation representation.name = 'rectangular array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_ context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit (rectangular_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition
rectangular_array_placement_ group_component to length_data_element (as terminus_row_positive_y_ displacement)	PATH			

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <-> property_definition.definition property_definition <- {property_definition property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context => representation_context => global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']}] representation.items[i] -> representation_item =>

Table 8 – Mapping table for assembly_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'terminus row positive y displacement'} measure_representation_item <= {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component </pre>
SUB_ASSEMBLY_RELATIONSHIP	(assembly_component_usage)	44		
sub_assembly_reference_designation	(assembly_component_usage.reference_designator) (shape_aspect_relationship.description)	44 41		
sub_assembly_relationship to assembly_component (as related_component)	PATH			<pre> (assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'}) product_definition_relationship.related_ product_definition -> product_definition => component_definition) </pre>
sub_assembly_relationship to assembly_group_component (as relating_component)	PATH		338	<pre> (assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'}) product_definition_relationship.related_ product_definition -> </pre>

Table 8 – Mapping table for assembly_component_grouping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				product_definition => component_definition => assembly_group_component_definition)

Table 9 – Mapping table for assembly_component_placement_requirements UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_GROUP_AS-SIGNMENT	(product_definition_relationship)	41		{product_definition_relationship.product_definition_relationship.member'}}
component_group_assignment to assembly_component (as assigned_component)	PATH		265, 266, 338, 333	(product_definition_relationship.product_definition_relationship.related_product_definition -> product_definition => component_definition)
component_group_assignment to placement_group (as associated_group)	PATH		265, 266, 338, 333	(product_definition_relationship.product_definition_relationship.related_product_definition -> product_definition => component_definition => group_product_definition)
COMPONENT_PLACEMENT_RESTRICTION_ASSIGNMENT	representation	43		representation {representation.name = 'component placement restriction assignment'}
components_permitted	descriptive_representation_item	45	109	representation.items[i] -> {representation_item representation_item.name = 'components permitted'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'maximum mounting clearance'}
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_negative_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation representation {representation.name = 'maximum negative component height'}
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_positive_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'maximum positive component height'}
component_placement_- restriction_assignment to datum_based_length_measure (as minimum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'minimum mounting clearance'}
component_placement_- restriction_assignment to item_restricted_requirement_- occurrence (as requirement)	PATH		42, 340	representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition =>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> requirements_property => grouped_requirements_property {[grouped_requirements_property <= group group.name = 'item restricted requirements property']} </pre>
component_placement_- restriction_assignment to mounting_restriction_area (as area)	PATH		42, 340, 333	<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement restriction assignment area'} representation_relationship.rep_2 -> representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_area </pre>
component_placement_- restriction_assignment to mounting_restriction_volume (as volume)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement restriction assignment volume'} representation_relationship.rep_2 -> </pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_volume </pre>
INTERCONNECT_MODULE_- COMPONENT_SURFACE_FEA- TURE	interconnect_module_- component_surface_feature	210	333	<pre> interconnect_module_component_surface_feature <= shape_aspect {shape_aspect.description = 'interconnect module component surface feature'} interconnect_module_component_surface_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface')} </pre>
interconnect_module_- component_surface_feature to interconnect_module_surface_- feature (as definition)	PATH			

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}</pre>
ITEM_RESTRICTED_REQUIREMENT_OCCURRENCE	grouped_requirements_property	210		<pre>grouped_requirements_property <= [group {group group.name = 'item restricted requirements property'}} [requirements_property <= property_definition] grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = representation {representation.name = 'component external reference'} representation</pre>
item_restricted_requirement_- occurrence to component_external_reference (as basis)	PATH			
item_restricted_requirement_- occurrence to component_feature_external_- reference (as basis)	PATH			<pre>grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment</pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = representation_item {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item </pre>
item_restricted_requirement_- occurrence to ee_requirement_occurrence (as basis)	PATH			<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = requirements_property requirements_property </pre>
item_restricted_requirement_- occurrence to assembly_component (as basis)	PATH		333, 340, 265, 266, 338	<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = component_definition) grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => </pre>
item_restricted_requirement_- occurrence to component_feature (as basis)	PATH		333	<pre> applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = component_terminal) (group_assigned_item = component_interface_- terminal) </pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(group_assigned_item = laminate_component_- interface_terminal) (group_assigned_item = component_feature) grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment =></pre>
item_restricted_requirement_- occurrence to part_feature (as basis)	PATH			<pre>applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = shape_aspect shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']} (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature)</pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
LAYOUT_SPACING_REQUIREMENT_OCCURRENCE	grouped_requirements_property	210	203	grouped_requirements_property <= [requirements_property] [{group group.name = 'layout spacing requirements property'}] grouped_requirements_property <= group {(group.description = 'nearest boundary') (group.description = 'centroid') (group.description = 'furthest boundary')}} group.description grouped_requirements_property <= group <-
of_spacing_type	PATH			
dependent_design_object_category	characterized_object.name	210	340	

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = design_object design_object <= characterized_object {characterized_object.description = 'dependent design object category'} characterized_object.name {(characterized_object.name = 'assembly module assembly component') (characterized_object.name = 'assembly module component terminal') (characterized_object.name = 'assembly module component') (characterized_object.name = 'assembly module macro component join terminal') (characterized_object.name = 'assembly module macro component') (characterized_object.name = 'assembly module') (characterized_object.name = 'bare die component terminal') (characterized_object.name = 'bare die component') (characterized_object.name = 'bonded conductive base blind via') (characterized_object.name = 'buried via') (characterized_object.name = 'cable component') (characterized_object.name = 'component feature') </pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'component mounting feature') (characterized_object.name = 'component termination passage and contact size dependent land') (characterized_object.name = 'component termination passage and contact size dependent non functional land') (characterized_object.name = 'component termination passage dependent land') (characterized_object.name = 'component termination passage dependent non functional land') (characterized_object.name = 'component termination passage') (characterized_object.name = 'conductive interconnect element with pre defined transitions') (characterized_object.name = 'conductive interconnect element with user defined single transition') (characterized_object.name = 'connected area component') (characterized_object.name = 'contact size dependent land') (characterized_object.name = 'contact size dependent non functional land') (characterized_object.name = 'cutout edge segment') (characterized_object.name = 'cutout') (characterized_object.name = 'dielectric material passage') (characterized_object.name = 'electrical isolation laminate component') (characterized_object.name = 'embedded component terminal')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'fiducial') (characterized_object.name = 'fill area') (characterized_object.name = 'filled area material removal laminate component') (characterized_object.name = 'interconnect component interface terminal') (characterized_object.name = 'interconnect component join terminal') (characterized_object.name = 'interconnect module assembly component') (characterized_object.name = 'interconnect module component surface feature') (characterized_object.name = 'interconnect module component terminal') (characterized_object.name = 'interconnect module component') (characterized_object.name = 'interconnect module edge segment') (characterized_object.name = 'interconnect module edge') (characterized_object.name = 'interconnect module macro component join terminal') (characterized_object.name = 'interconnect module macro component') (characterized_object.name = 'interface access material removal laminate component') (characterized_object.name = 'interface component') (characterized_object.name = 'interfacial connection') (characterized_object.name = 'internal probe access area') (characterized_object.name = 'join two physical connectivity definition supporting printed component')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'laminated component') (characterized_object.name = 'laminated text component') (characterized_object.name = 'laminated text string component') (characterized_object.name = 'land') (characterized_object.name = 'material removal laminate component') (characterized_object.name = 'minimally defined component terminal') (characterized_object.name = 'movable packaged component join terminal') (characterized_object.name = 'multi layer material removal laminate component') (characterized_object.name = 'non conductive base blind via') (characterized_object.name = 'non functional land') (characterized_object.name = 'packaged component join terminal') (characterized_object.name = 'packaged component') (characterized_object.name = 'packaged connector component interface terminal') (characterized_object.name = 'packaged connector component') (characterized_object.name = 'partially plated cutout') (characterized_object.name = 'partially plated interconnect module edge') (characterized_object.name = 'physical component') (characterized_object.name = 'physical connectivity interrupting cutout') (characterized_object.name = 'physical laminate component')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'plated conductive base blind via') (characterized_object.name = 'plated cutout edge segment') (characterized_object.name = 'plated cutout') (characterized_object.name = 'plated interconnect module edge segment') (characterized_object.name = 'plated interconnect module edge') (characterized_object.name = 'plated passage or unsupported passage') (characterized_object.name = 'plated passage') (characterized_object.name = 'probe access area') (characterized_object.name = 'routed interconnect component') (characterized_object.name = 'routed physical component') (characterized_object.name = 'special symbol laminate component') (characterized_object.name = 'stratum feature template component') (characterized_object.name = 'stratum feature') (characterized_object.name = 'thermal isolation laminate component') (characterized_object.name = 'unrouted conductive interconnect element') (characterized_object.name = 'unsupported passage dependent non functional land') (characterized_object.name = 'unsupported passage') (characterized_object.name = 'via and contact size dependent land') (characterized_object.name = 'via and contact size dependent non functional land')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'via dependent land') (characterized_object.name = 'via dependent non functional land') (characterized_object.name = 'via')
reference_design_object_category	characterized_object.name	210		grouped_requirements_property <= group <= group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = design_object design_object <= characterized_object {characterized_object characterized_object.description = 'reference design object category'} characterized_object.name {(characterized_object.name = 'assembly module assembly component') (characterized_object.name = 'assembly module component terminal') (characterized_object.name = 'assembly module component') (characterized_object.name = 'assembly module macro component join terminal') (characterized_object.name = 'assembly module macro component') (characterized_object.name = 'assembly module') (characterized_object.name = 'bare die component terminal') (characterized_object.name = 'bare die component')}

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'bonded conductive base blind via') (characterized_object.name = 'buried via') (characterized_object.name = 'cable component') (characterized_object.name = 'component feature') (characterized_object.name = 'component mounting feature') (characterized_object.name = 'component termination passage and contact size dependent land') (characterized_object.name = 'component termination passage and contact size dependent non functional land') (characterized_object.name = 'component termination passage dependent land') (characterized_object.name = 'component termination passage dependent non functional land') (characterized_object.name = 'component termination passage') (characterized_object.name = 'conductive interconnect element with pre defined transitions') (characterized_object.name = 'conductive interconnect element with user defined single transition') (characterized_object.name = 'connected area component') (characterized_object.name = 'contact size dependent land') (characterized_object.name = 'contact size dependent non functional land') (characterized_object.name = 'cutout edge segment')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'cutout') (characterized_object.name = 'dielectric material passage') (characterized_object.name = 'electrical isolation laminate component') (characterized_object.name = 'embedded component terminal') (characterized_object.name = 'fiducial') (characterized_object.name = 'fill area') (characterized_object.name = 'filled area material removal laminate component') (characterized_object.name = 'interconnect component interface terminal') (characterized_object.name = 'interconnect component join terminal') (characterized_object.name = 'interconnect module assembly component') (characterized_object.name = 'interconnect module component surface feature') (characterized_object.name = 'interconnect module component terminal') (characterized_object.name = 'interconnect module component') (characterized_object.name = 'interconnect module edge segment') (characterized_object.name = 'interconnect module edge') (characterized_object.name = 'interconnect module macro component join terminal') (characterized_object.name = 'interconnect module macro component') (characterized_object.name = 'interface access material removal laminate component') (characterized_object.name = 'interface component')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'interfacial connection') (characterized_object.name = 'internal probe access area') (characterized_object.name = 'join two physical connectivity definition supporting printed component') (characterized_object.name = 'laminated component') (characterized_object.name = 'laminated text component') (characterized_object.name = 'laminated text string component') (characterized_object.name = 'land') (characterized_object.name = 'material removal laminate component') (characterized_object.name = 'minimally defined component terminal') (characterized_object.name = 'movable packaged component join terminal') (characterized_object.name = 'multi layer material removal laminate component') (characterized_object.name = 'non conductive base blind via') (characterized_object.name = 'non functional land') (characterized_object.name = 'packaged component join terminal') (characterized_object.name = 'packaged component') (characterized_object.name = 'packaged connector component interface terminal') (characterized_object.name = 'packaged connector component') (characterized_object.name = 'partially plated cutout')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'partially plated interconnect module edge') (characterized_object.name = 'physical component') (characterized_object.name = 'physical connectivity interrupting cutout') (characterized_object.name = 'physical laminated component') (characterized_object.name = 'plated conductive base blind via') (characterized_object.name = 'plated cutout edge segment') (characterized_object.name = 'plated cutout') (characterized_object.name = 'plated interconnect module edge segment') (characterized_object.name = 'plated interconnect module edge') (characterized_object.name = 'plated passage or unsupported passage') (characterized_object.name = 'plated passage') (characterized_object.name = 'probe access area') (characterized_object.name = 'routed interconnect component') (characterized_object.name = 'routed physical component') (characterized_object.name = 'special symbol laminated component') (characterized_object.name = 'stratum feature template component') (characterized_object.name = 'stratum feature') (characterized_object.name = 'thermal isolation laminated component') (characterized_object.name = 'unrouted conductive interconnect element')

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'unsupported passage dependent non functional land') (characterized_object.name = 'unsupported passage') (characterized_object.name = 'via and contact size dependent land') (characterized_object.name = 'via and contact size dependent non functional land') (characterized_object.name = 'via dependent land') (characterized_object.name = 'via dependent non functional land') (characterized_object.name = 'via')}}
layout_spacing_requirement_- occurrence to tolerance_characteristic (as required_coordinated_- characteristic)	PATH			requirements_property <= property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item {representation_item (representation_item.name = 'tolerance') (representation_item.name = 'minimum tolerance') (representation_item.name = 'maximum tolerance') (representation_item.name = 'plus minus tolerance') (representation_item.name = 'symmetrical tolerance') (representation_item.name = 'statistical tolerance')}}

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
MOUNTING_RESTRICTION_- mounting_restriction_area to interconnect_module_- component_surface_feature (as mounting_surface)	mounting_restriction_area PATH	210	333	<pre> mounting_restriction_area <= shape_aspect mounting_restriction_area <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mounting surface'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect.description = 'interconnect module component surface feature'} shape_aspect => interconnect_module_component_surface_feature </pre>
mounting_restriction_area to assembly_module (as scope)	PATH			<pre> mounting_restriction_area <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']] </pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit
MOUNTING_RESTRICTION_- VOLUME	mounting_restriction_volume	210		mounting_restriction_volume <= shape_aspect
mounting_restriction_volume to interconnect_module_- component_surface_feature (as mounting_surface)	PATH			mounting_restriction_volume <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mounting surface'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect.description = 'interconnect module component surface feature'} shape_aspect => interconnect_module_component_surface_feature
mounting_restriction_volume to assembly_module (as scope)	PATH			mounting_restriction_volume <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}]}

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit
PLACEMENT_GROUP	(group_product_definition)	210	265, 266, 338, 333, 340	(group_product_definition <= component_definition <= product_definition {product_definition [product_definition.description = 'placement group'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement']})
design_specific_usage	product.name	41	267, 268, 263, 264	(group_product_definition <= component_definition <=) product_definition product_definition.format -> product_definition.format product_definition.format.of_product -> product product.name
placement_group to ee_requirement_occurrence (as requirement)	PATH		265, 266, 338, 333, 340	(group_product_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition) characterized_definition <= property_definition.definition {property_definition property_definition.name = 'requirement'} property_definition <= property_definition.relationship.related_ property_definition

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{property_definition_relationship property_definition_relationship.name = 'requirement'} property_definition_relationship.relatin_ - property_definition -> property_definition => requirements_property</pre>
PLACEMENT_GROUP_AREA_- ASSIGNMENT	shape_aspect_relationship	41	245	<pre>{shape_aspect_relationship shape_aspect_relationship.name = 'area impacted group'}</pre>
placement_group_area_- assignment to mounting_restriction_area (as area)	PATH		333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => mounting_restriction_area</pre>
placement_group_area_- assignment to placement_group (as placed_group)	PATH		340, 265, 266, 338	<pre>shape_aspect_relationship shape_aspect_relationship.relatin_shape_- aspect -> (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'design requirement'}</pre>

Table 9 – Mapping table for assembly_component_placement_requirements UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				product_definition => component_definition => group_product_definition)
PLACEMENT_GROUP_VOLUME_ASSIGNMENT	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'volume impacted_group'}
placement_group_volume_- assignment to mounting_restriction_area (as volume)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => mounting_restriction_volume
placement_group_volume_- assignment to placement_group (as placed_group)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement'} product_definition => component_definition => group_product_definition)

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_FEATURE_TO_- PHYSICAL_USAGE_VIEW_AS- SIGNMENT	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'component feature to physical usage assignment'}
component_feature_to_physical_- usage_view_assignment to component_feature (as assigned_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature)
component_feature_to_physical_- usage_view_assignment to part_feature (as defined_part_feature)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface')

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
COMPONENT_TERMINAL_- TO_ASSEMBLY_MODULE_- INTERFACE_TERMINAL_AS- SIGNMENT	shape_aspect_relationship	41		<pre> {shape_aspect_relationship [shape_aspect_relationship.name = 'component feature to physical usage assignment'] [shape_aspect_relationship.description = 'component terminal to assembly module interface terminal assignment']] </pre>

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- assembly_module_interface_- terminal_assignment to assembly_module_component_- terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'assembly module component terminal'} shape_aspect => component_terminal
component_terminal_to_- assembly_module_interface_- terminal_assignment to assembly_module_interface_- terminal (as usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'assembly module'} shape_aspect => assembly_module_terminal => assembly_module_interface_terminal
component_terminal_to_- assembly_module_interface_- terminal_assignment to component_interface_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'interconnect component interface terminal') (shape_aspect.description = 'packaged connector component interface terminal')}} shape_aspect => component_interface_terminal

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- assembly_module_interface_- terminal_assignment to interconnect_module_- component_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'} shape_aspect => component_terminal
COMPONENT_TERMINAL_- TO_ASSEMBLY_MODULE_- JOIN_TERMINAL_ASSIGN- MENT	shape_aspect_relationship	41		{shape_aspect_relationship [shape_aspect_relationship.name = 'component feature to physical usage assignment'] [shape_aspect_relationship.description = 'component terminal to assembly module join terminal assignment']}]
component_terminal_to_- assembly_module_join_terminal_- assignment to assembly_module_macro_- component_join_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'assembly module component terminal'} shape_aspect => component_terminal => assembly_module_macro_component_join_terminal
component_terminal_to_- assembly_module_join_terminal_- assignment to assembly_module_join_terminal (as usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.name = 'assembly module' shape_aspect => assembly_module_terminal => assembly_module_join_terminal
component_terminal_to_ assembly_module_join_terminal_ assignment to interconnect_module_macro_ component_join_terminal (as assigned_terminal)	PATH			shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'} shape_aspect => component_terminal => interconnect_module_macro_component_join_ terminal
COMPONENT_TERMINAL_ TO_PCA_TERMINAL_ASSIGN- MENT	shape_aspect_relationship	41	52	{shape_aspect_relationship [shape_aspect_relationship.name = 'component feature to physical usage assignment'] [shape_aspect_relationship.description = 'component terminal to pca terminal assignment']}
component_terminal_to_pca_ terminal_assignment to assembly_module_component_ terminal (as assigned_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'assembly module component terminal'} shape_aspect => component_terminal

Table 10 – Mapping table for assembly_design_information_assignment_to_usage_view UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_pca_- terminal_assignment to component_interface_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'interconnect component interface terminal') (shape_aspect.description = 'packaged connector component interface terminal')}} shape_aspect => component_interface_terminal
component_terminal_to_pca_- terminal_assignment to interconnect_module_- component_terminal (as assigned_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'} shape_aspect => component_terminal
component_terminal_to_pca_- terminal_assignment to pca_terminal (as usage_view_terminal)	PATH		51	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'pca terminal'} shape_aspect => assembly_module_terminal

Table 11 – Mapping table for assembly_functional_interface_requirement UoF

Application element	AIM element	Source	Rules	Reference path
MATING_CONNECTOR_TERMINATION	shape_aspect	41	209, 210	{shape_aspect shape_aspect.description = 'mating connector termination'}
name	shape_aspect.name	41		
mating_connector_termination to ee_requirement_occurrence (as signal_definition)	PATH		340	shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'signal definition'} property_definition <- property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'signal_definition'} property_definition.relationship.relatin_ property_definition -> property_definition => requirements_property
mating_connector_termination to mating_connector_component (as accessed_connector)	PATH		340, 338	shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 11 – Mapping table for assembly_functional_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.description = 'mating connector'} product_definition => component_definition </pre>
mating_connector_termination to packaged_part_terminal (as definition)	PATH		333	<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated terminal'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => packaged_part_terminal </pre>
TERMINATION_CONSTRAINT	group_shape_aspect	210	333, 340, 265, 266, 338	<pre> group_shape_aspect <= shape_aspect {shape_aspect.description = 'termination constraint'} {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => component_definition => group_product_definition} </pre>

Table 11 – Mapping table for assembly_functional_interface_requirement UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
termination_constraint_to_ee_requirement_occurrence (as termination_usage_constraint)	PATH		333, 340	<pre> group_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'termination usage constraint'} property_definition <- property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'termination usage constraint'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property </pre>
termination_constraint_to_mating_connector_termination (as constrained_termination)	PATH			<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constrained termination member'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'mating connector termination'} </pre>

Table 12 – Mapping table for assembly_functional_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect {shape_aspect.description = 'assembly module component_terminal' (shape_aspect.description = 'bare die component_terminal' (shape_aspect.description = 'interconnect component join terminal' (shape_aspect.description = 'interconnect module component_terminal' (shape_aspect.description = 'minimally defined component_terminal' (shape_aspect.description = 'packaged component join terminal'}}} (component_interface_terminal <= shape_aspect)
PHYSICAL_NODE_REQUIREMENT_TO_IMPLEMENTING_COMPONENT_ALLOCATION	physical_node_requirement_to_implementing_component_allocation	210		{[physical_node_requirement_to_implementing_ component_allocation <= shape_aspect] [physical_node_requirement_to_implementing_ component_allocation <= shape_aspect_relationship]}
physical_node_requirement_to_implementing_component_allocation to assembly_component (as implementation_component)	PATH		338, 340, 333	physical_node_requirement_to_implementing_ component_allocation <= shape_aspect_relationship.related_shape_aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition

Table 12 – Mapping table for assembly_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition {[product_definition => component_definition] [product_definition.description != 'laminated component']}]
physical_node_requirement_to_implementing_component_allocation_to assembly_joint (as implementation_joint_set)	PATH			physical_node_requirement_to_implementing_component_allocation <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'implementation_joint'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint
physical_node_requirement_to_implementing_component_allocation_to interface_mounted_join (as implementation_interface_joint_set)	PATH			physical_node_requirement_to_implementing_component_allocation <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'implementation_interface_joint'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => interface_mounted_join

Table 12 – Mapping table for assembly_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_node_requirement_to_implementation_component_allocation to physical_connectivity_definition (as physical_node_requirement)	PATH			physical_node_requirement_to_implementation_component_allocation <= shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => physical_connectivity_definition
TEST_REQUIREMENT_ALLOCATION	requirement_allocation_group	210		{[requirement_allocation_group <= property_definition_relationship][property_definition_relationship property_definition_relationship.name = 'test requirement_allocation'] [requirement_allocation_group <= group]} requirement_allocation_group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = product_definition_formation}
test_requirement_allocation to ee_product_version (as target_product_object)	PATH			requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment group_assigned_item = product_definition_formation
test_requirement_allocation to test_requirement_occurrence (as allocated_requirement)	PATH			requirement_allocation_group <= property_definition_relationship property_definition_relationship.relatng_shape_aspect -> property_definition => {property_definition.description = 'test requirements_property requirements_property

Table 12 – Mapping table for assembly_functional_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
test_requirement_allocation to part_feature_model_port_- assignment (as analytical_model_based_test_- access)	PATH			<pre> requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> property_definition_representation {property_definition_representation property_definition_representation.name = 'part feature model port assignment'} </pre>
test_requirement_allocation to functional_usage_view_to_part_- terminal_assignment (as functional_specification_- based_test_access)	PATH			<pre> requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal allocation'} </pre>

Table 13 – Mapping table for assembly_interface_component UoF

Application element	AIM element	Source	Rules	Reference path
EE_SPECIFICATION	ee_specification	210		ee_specification <= document
ee_specification to organization (as source)	PATH			<pre> ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_role organization_role.name = 'document source'} organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} </pre>

Table 13 – Mapping table for assembly_interface_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
INTERFACE_COMPONENT	interface_component_definition	210	338, 333, 340	<pre> interface_component_definition <= component_definition <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'} {product_definition <- product_definition.relationship.related_- product_definition {product_definition.relationship product_definition.relationship.name = 'instantiated part'}} product_definition.relationship product_definition.relationship.relatiing_- product_definition -> [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'] [product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)]} </pre>
PACKAGED_CONNECTOR_- COMPONENT	packaged_component	210	338	<pre> packaged_component <= component_definition <= product_definition </pre>

Table 13 – Mapping table for assembly_interface_component UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>{[product_definition.description = 'packaged connector component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']]}</pre>
packaged_connector_component to packaged_connector (as usage_definition)	PATH		338	<pre>packaged_component <= component_definition <= product_definition <- product_definition_relationship.related_- product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship.relatin_- product_definition -> product_definition => physical_unit => (packaged_part => packaged_connector) (externally_defined_physical_unit => externally_defined_packaged_part => externally_defined_packaged_connector) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector)</pre>

Table 14 – Mapping table for assembly_marking UoF

Application element	AIM element	Source	Rules	Reference path
CONFIGURATION_CONTROLLED_MESSAGE	descriptive_representation_item	45	60	
message_content	descriptive_representation_item.description	45		
message_type_specification	representation_item.name	43	109	descriptive_representation_item <= representation_item representation_item.name {(representation_item.name = 'approval date') (representation_item.name = 'manufacturer code') (representation_item.name = 'part number') (representation_item.name = 'revision identifier')}}
configuration_controlled_message to ee_approval (as message_source)	PATH		13	descriptive_representation_item <= representation_item <- representation.items[i] representation approval_assigned_item = representation approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
configuration_controlled_message to ee_product_version (as message_source)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_representation property_definition_representation

Table 14 – Mapping table for assembly_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.definition -> product_definition_formation descriptive_representation_item <= representation_item <- representation.items[i] representation organization_assigned_item = representation organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization </pre>
configuration_controlled_message to organization (as message_source)	PATH			
EE_SPECIFICATION	ee_specification	210		<pre> ee_specification <= document </pre>
ee_specification to organization (as source)	PATH			<pre> ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment </pre>

Table 14 – Mapping table for assembly_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> organization_assignment.role -> organization_role organization_role.name = 'document source' organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
MARKING	representation	43	206	<pre> {representation representation.name = 'marking'} representation <- [property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 14 – Mapping table for assembly_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect[]
marking to axis_placement (as placement)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'marking location'} representation_item => geometric_representation_item => placement
marking to ee_text (as marked_message)	PATH			representation representation.items[i] -> representation_item => descriptive_representation_item
marking to ee_material (as marking_material)	PATH			representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> {characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect} characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_ designation

Table 14 – Mapping table for assembly_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'marking material'} representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document <- document_usage_constraint.source document_usage_constraint document_usage_constraint.subject_element = 'marking specification'} document => ee_specification </pre>
marking to physical_unit_3d_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> representation {representation.name = '3d bound volume shape'} </pre>
marking to physical_unit_planar_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> representation {representation.name = '3d bound volume shape'} </pre>

Table 14 – Mapping table for assembly_marking UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation {representation.name = 'planar projected shape'}
marking to feature_shape_definition (as message_representation)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'}
marking to styled_text (as message_representation_text)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'} representation_items[i] -> representation_item representation_item.name = 'marking text' representation_item => (geometric_representation_item => text_literal) (styled_item => annotation_occurrence => annotation_text_occurrence)

Table 15 – Mapping table for assembly_module_design UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLED_WITH_BONDING	assembly_joint	210	344, 340, 339, 354	[assembly_joint <= shape_aspect_relationship shape_aspect_relationship.name = 'assembled with bonding'] [shape_aspect]
assembled_with_bonding to assembly_bond_definition (as default_bond_definition)	PATH		333	assembly_joint <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'default definition usage'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => assembly_bond_definition
ASSEMBLED_WITH_FASTENERS	assembly_joint	210	344	assembly_joint <= [shape_aspect_relationship shape_aspect_relationship.name = 'assembled with fasteners'] [shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => assembly_group_component_definition]
assembled_with_fasteners to assembly_group_component (as associated_assembly)	PATH			assembly_joint <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => assembly_group_component_definition
ASSEMBLY_COMPONENT	(component_definition)	210	333, 340, 338	(component_definition <= (product_definition {[product_definition product_definition.description != 'printed component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']}))
assembly_component to physical_unit (as usage_definition)	PATH		338	component_definition <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.relatiing_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit) </pre>
ASSEMBLY_COMPOSITION_- RELATIONSHIP	assembly_component_usage	44	354, 339, 15	<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly composition'} </pre>
assembly_composition_- relationship to assembly_component (as component)	PATH		354, 339, 338, 340, 333	<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{product_definition => component_definition})
assembly_composition_- relationship to ee_material (as assembly_bonding_agent)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= material_designation.definitions[i] material_designation
assembly_composition_- relationship to physical_unit_design_view (as assembly)	PATH		354, 339, 338	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
assembly_composition_- relationship to security_classification (as security_code)	PATH			assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <= applied_security_classification_- assignment.items[i] applied_security_classification_assignment <=

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification
ASSEMBLY_JOINT	assembly_joint	210		assembly_joint <= [shape_aspect_relationship] [shape_aspect]
description	shape_aspect_ relationship.description	41	344	assembly_joint <= shape_aspect_relationship shape_aspect_relationship.description
name	shape_aspect.name	41	333	assembly_joint <= shape_aspect shape_aspect.name
assembly_joint to assembly_group_component (as associated_assembly)	PATH			assembly_joint <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => assembly_group_component_definition
assembly_joint to assembly_module (as associated_assembly)	PATH			assembly_joint <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition {product_definition.name = 'assembly module'}
assembly_joint to assembly_- module_component_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'assembly module component terminal'} shape_aspect => component_terminal
assembly_joint to assembly_- module_component_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'assembly module component terminal'} shape_aspect => component_terminal
assembly_joint to bare_die_component_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'bare die component terminal'} shape_aspect => component_terminal
assembly_joint to bare_die_component_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'bare die component terminal'} shape_aspect => component_terminal

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
assembly_joint to component_mounting_feature (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> shape_aspect => component_mounting_feature
assembly_joint to component_mounting_feature (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_mounting_feature
assembly_joint to interconnect_- component_interface_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect component interface terminal'} shape_aspect => component_interface_terminal
assembly_joint to interconnect_- component_interface_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect component interface terminal'} shape_aspect => component_interface_terminal
assembly_joint to interconnect_- component_join_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect component join terminal'}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
assembly_joint to interconnect_ component_join_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect component join terminal'} shape_aspect => component_terminal
assembly_joint to interconnect_module_ component_surface_feature (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => interconnect_module_component_surface_feature
assembly_joint to interconnect_module_ component_surface_feature (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => interconnect_module_component_surface_feature
assembly_joint to interconnect_ module_component_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'} shape_aspect => component_terminal
assembly_joint to interconnect_ module_component_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
assembly_joint to package_terminal_occurrence (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'package terminal occurrence'} shape_aspect => component_terminal
assembly_joint to package_terminal_occurrence (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'package terminal occurrence'} shape_aspect => component_terminal
assembly_joint to packaged_- component_join_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relate_shape_- aspect -> {shape_aspect shape_aspect.description = 'packaged component join terminal'} shape_aspect => component_terminal
assembly_joint to packaged_- component_join_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'packaged component join terminal'}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
assembly_joint to packaged_connector_component_- interface_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> {shape_aspect shape_aspect.description = 'packaged connector component interface terminal'} shape_aspect => component_interface_terminal
assembly_joint to packaged_connector_component_- interface_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'packaged connector component interface terminal'} shape_aspect => component_interface_terminal
ASSEMBLY_MATERIAL_COM- POSITION_RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44		(next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}) ([quantified_assembly_component_usage] [next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}])

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
as_required	PATH	41		quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit measure_with_unit.value_component -> measure_value measure_value = descriptive_measure {(descriptive_measure = 'as_required') (descriptive_measure = 'quantity specified')}]
assembly_material_composition_- relationship to data_element (as quantity)	PATH			quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit
assembly_material_composition_- relationship to material_definition (as component_material)	PATH			next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'material definition'}
assembly_material_composition_- relationship to physical_unit_design_view (as assembly)	PATH		338	next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition ->

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition product_definition => physical_unit
assembly_material_composition_- relationship to security_classification (as security_code)	PATH			next_assembly_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification
ASSEMBLY_MODULE	physical_unit	210	268	physical_unit <= product_definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']}]}
ASSEMBLY_MODULE_COM- PONENT_TERMINAL	component_terminal	210	333	component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'assembly module component terminal'] [shape_aspect.product_definitional = .TRUE.]}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_module_component_- terminal to assembly_module_component (as associated_component)	PATH		333, 340, 268, 338	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']} product_definition => component_definition </pre>
assembly_module_component_- terminal to assembly_module_terminal (as definition)	PATH		333, 340, 268	<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition product_definition.name = 'assembly module' shape_aspect => assembly_module_terminal
ASSEMBLY_MODULE_- MACRO_COMPONENT	component_definition	210		component_definition <= product_definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'macro occurrence']}
assembly_module_macro_- component to assembly_module (as design_definition)	PATH			component_definition <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'design definition'} product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
ASSEMBLY_MODULE_- MACRO_COMPONENT_JOIN_- TERMINAL	assembly_module_macro_- component_join_terminal	210		<pre> assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'assembly module component terminal'] [shape_aspect.product_definitional = .TRUE.]} </pre>
assembly_module_macro_- component_join_terminal to assembly_module_macro_- component (as associated_component)	PATH			<pre> assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'macro occurrence']} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => component_definition
assembly_module_macro_- component_join_terminal to assembly_module_join_terminal (as definition)	PATH			assembly_module_macro_component_join_terminal <= component_terminal <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'assembly module'} shape_aspect => assembly_module_terminal => assembly_module_join_terminal
BARE_DIE_COMPONENT_- TERMINAL	component_terminal	210	333	component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'bare die component terminal'] [shape_aspect.product_definitional = .TRUE.]}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
bare_die_component_terminal to bare_die_component (as associated_component)	PATH		333, 340, 338	component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.description = 'bare die component'} product_definition => component_definition
bare_die_component_terminal to bare_die_terminal (as definition)	PATH		333	component_terminal <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship.related_shape_ aspect -> shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal
COMPONENT_FEATURE	(component_terminal) (component_feature) (laminar_component_interface_ terminal) (interconnect_module_ component_surface_feature)	210 210 210 210 210	333, 37	(component_terminal <= shape_aspect {shape_aspect [shape_aspect.product_definitional = .TRUE.] [(shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal')]

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') (shape_aspect.description = 'packaged component join terminal') }} (component_interface_terminal <= shape_aspect) (component_feature <= shape_aspect) (laminate_component_interface_terminal <= shape_aspect) (interconnect_module_component_surface_feature <= shape_aspect)</pre>
component feature to assembly_component (as associated_component)	PATH		340, 338, 333	<pre>((component_terminal <=) (component_interface_terminal <=) (component_feature <=) (laminate_component_interface_terminal <=) (interconnect_module_component_surface_feature <=) shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition)</pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> ((component_terminal <=) (laminate_component_interface_terminal <=) shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminate component'] [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect) </pre>
component_feature to component_termination_passage_- template_terminal (as definition)	PATH		333	<pre> (component_terminal <=) (laminate_component_interface_terminal <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'instantiated feature' shape_aspect_relationship shape_aspect_relationship.relying_shape_ aspect -> shape_aspect {shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal') } (component_terminal <=) (component_interface_terminal <=) (component_feature <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature' shape_aspect_relationship shape_aspect_relationship.relying_shape_ aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') </pre>
component_feature to part_feature (as definition)	PATH		333	

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal)

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => package_body_surface)}
COMPONENT_FEATURE_- MAKE_FROM_RELATIONSHIP	make_from_feature_relationship	210		make_from_feature_relationship <= shape_aspect_relationship
component_feature_make_from_- relationship to component_feature (as resultant_feature)	PATH		344, 333	make_from_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
component_feature_make_from_- relationship to component_feature (as reusable_feature)	PATH			make_from_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.relation_shape_- aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
COMPONENT_FEATURE_RE- LATIONSHIP	component_feature_relationship	210		{[component_feature_relationship <= shape_aspect] [component_feature_relationship <= shape_aspect_relationship]}
component_feature_relationship to component_feature (as related_component_feature)	PATH		333	component_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => (component_terminal))

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
component_feature_relationship to component_feature (as relating_component_feature)	PATH		333	component_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
COMPONENT_INTERFACE_- TERMINAL	component_interface_terminal	210	333	component_interface_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect component_interface_terminal') (shape_aspect.description = 'packaged connector_component_interface_terminal')}} component_interface_terminal <=
component_interface_terminal to physical_component (as associated_component)	PATH			shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition {product_definition product_definition.description != 'laminated component'}}

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence' product_definition => component_definition
COMPONENT_MAKE_FROM_- RELATIONSHIP	(make_from_usage_option) (laminar_group_component_- make_from_relationship)	44 210		(make_from_usage_option) (laminar_group_component_make_from_- relationship <= make_from_usage_option)
component_make_from_- relationship to assembly_component (as resultant_component)	PATH			(make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition) (laminar_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => assembly_group_component_definition)
component_make_from_- relationship to assembly_component (as reusable_component)	PATH			(make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => product_definition =>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> component_definition (laminate_group_component_make_from_ - relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatin_ - product_definition -> product_definition => component_definition => assembly_group_component_definition) </pre>
COMPONENT_MATING_CONSTRAINT_CONDITION	component_mating_constraint_ - condition	210		<pre> component_mating_constraint_condition <= shape_aspect </pre>
constrained_degree_of_freedom	representation_item.name	43		<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation property_definition_representation.description = 'constrained degrees of freedom'} property_definition_representation.used_ - representation -> representation {representation representation.name = 'degrees of freedom'} representation.items[i] -> {(representation_item.name = 'x')} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(representation_item.name = 'y') (representation_item.name = 'z') (representation_item.name = 'u') (representation_item.name = 'v') (representation_item.name = 'w')}</pre>
name	shape_aspect.name	41		<pre>component_mating_constraint_condition <= shape_aspect shape_aspect.name</pre>
component_mating_constraint_- condition to assembly_joint (as constraining_joint)	PATH			<pre>component_mating_constraint_condition <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraining_joint'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint</pre>
component_mating_constraint_- condition to physical_unit_design_view (as of_context)	PATH			<pre>component_mating_constraint_condition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition => {product_definition product_definition.frame_of_reference -></pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'physical design'} physical_unit
COMPONENT_MOUNTING_- CLEARANCE_RELATIONSHIP	representation	43	38	{representation.name = 'mounting clearance'}
component_mounting_clearance_- relationship to length_data_element (as maximum_mounting_clearance)	PATH		109	{representation.name = 'mounting clearance'} representation representation.items[i] -> {representation_item representation_item.name = 'maximum height'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
component_mounting_clearance_- relationship to length_data_element (as minimum_mounting_clearance)	PATH			{representation.name = 'mounting clearance'} representation representation.items[i] -> {representation_item representation_item.name = 'minimum height'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
component_mounting_clearance_- relationship to component_feature_relationship (as feature_relationship)	PATH			{representation.name = 'mounting clearance'} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition property_definition.name = 'component mounting property'} property_definition.definition -> characterized_definition shape_definition = shape_aspect shape_aspect => component_feature_relationship </pre>
COMPONENT_MOUNTING_- FEATURE component_mounting_feature to part_mounting_feature (as definition)	component_mounting_feature PATH	210		<pre> component_mounting_feature <= shape_aspect component_mounting_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect => part_mounting_feature </pre>
COMPONENT_OVERLAP_RE- LATIONSHIP component_overlap_relationship to length_data_element (as maximum_added_clearance)	representation PATH	43 109	39	<pre> {representation.name = 'overlap clearance'} {representation.name = 'overlap clearance'} representation representation.items[i] -> {representation_item representation_item.name = 'maximum clearance'} representation_item => </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <= measure_with_unit => length_measure_with_unit
component_overlap_relationship to length_data_element (as minimum_added_clearance)	PATH		109	{representation.name = 'overlap clearance' representation representation.items[i] -> {representation_item representation_item.name = 'minimum clearance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
component_overlap_relationship to next_higher_assembly_- relationship (as current_component)	PATH		339, 354	{representation.name = 'overlap clearance'} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition [property_definition.name = 'component overlap property'] [property_definition.description = 'current component']} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => product_definition_usage =>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage])
component_overlap_relationship to next_higher_assembly_- relationship (as previously_placed_component)	PATH		339, 354	{representation.name = 'overlap clearance'} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition [property_definition.name = 'component overlap property'] [property_definition.description = 'previously placed component']} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage])
COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal) (component_feature) (laminar_component_interface_- terminal)	210 210 210 210	333	(component_terminal <= shape_aspect {shape_aspect [shape_aspect.product_definitional = .TRUE.] [(shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal')])

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal')]}) (component_interface_terminal <= shape_aspect) (component_feature <= shape_aspect) (laminate_component_interface_terminal <= shape_aspect) </pre>
global_swappable	descriptive_representation_item	45		<pre> component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'true')} descriptive_representation_item
local_swappable	descriptive_representation_item	45		component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')}} descriptive_representation_item
swap_code	descriptive_representation_ - item.description	45		component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation -> representation {representation.items[i] -> representation_item representation_item.name = 'swap code'} representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
CONNECTION_ZONE_- BASED_ASSEMBLY_JOINT	connection_zone_based_- assembly_joint	210		<pre> connection_zone_based_assembly_joint<= assembly_joint <= shape_aspect </pre>
connection_zone_based_- assembly_joint to connection_zone (as zone_1)	PATH			<pre> connection_zone_based_assembly_joint <= assembly_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 1'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
connection_zone_based_- assembly_joint to connection_zone (as zone_2)	PATH			<pre> connection_zone_based_assembly_joint <= assembly_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 2'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
INTERCONNECT_COMPO- NENT_INTERFACE_TERMI- NAL	component_interface_terminal	210	333, 189	<pre> component_interface_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interconnect component interface terminal'} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
name	shape_aspect.name	41		component_interface_terminal <= shape_aspect shape_aspect.name
interconnect_component_- interface_terminal to connection_zone (as connection)	PATH		333	component_interface_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
INTERCONNECT_COMPO- NENT_JOIN_TERMINAL	component_terminal	210	190	component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interconnect component join terminal'}
name	shape_aspect.name	41		component_terminal <= shape_aspect shape_aspect.name
interconnect_component_join_- terminal to physical_component (as associated_component)	PATH			component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_- definition

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => {[product_definition product_definition.description != 'laminated component']} [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']] component_definition </pre>
interconnect_component_join_- terminal to connection_zone (as connection)	PATH		333	<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'}} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}} </pre>
INTERCONNECT_MODULE_- COMPONENT_SURFACE_FEA- TURE	interconnect_module_- component_surface_feature	210	333	<pre> interconnect_module_component_surface_feature <= shape_aspect {shape_aspect.description = 'interconnect module component surface feature'}} interconnect_module_component_surface_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}} shape_aspect_relationship </pre>
interconnect_module_- component_surface_feature to interconnect_module_surface_- feature (as definition)	PATH			

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}} </pre>
INTERCONNECT_MODULE_- COMPONENT_TERMINAL	component_terminal	210	333	<pre> component_terminal <= shape_aspect {(shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'interconnect module stratum based terminal')}} </pre>
interconnect_module_- component_terminal to interconnect_module_component (as associated_component)	PATH		333, 340, 268, 338	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']] product_definition => component_definition</pre>
interconnect_module_- component_terminal to interconnect_module_terminal (as definition)	PATH		333, 340, 268	<pre>component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'interconnect module'} shape_aspect => interconnect_module_terminal</pre>
INTERCONNECT_MODULE_- MACRO_COMPONENT	component_definition	210		<pre>component_definition <= product_definition {product_definition [product_definition.name = 'interconnect module']}</pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'macro occurrence'.]}
interconnect_module_macro_- component to interconnect_module (as design_definition)	PATH			<pre> component_definition <= product_definition <- product_definition.relationship.related_- product_definition {product_definition.relationship product_definition.relationship.name = 'design definition'} product_definition.relationship product_definition.relationship.relating_- product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
INTERFACE_MOUNTED_JOIN	interface_mounted_join	210		<pre> interface_mounted_join <= [shape_aspect_relationship] [shape_aspect] </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interface_mounted_join to packaged_connector_component_- interface_terminal (as connection_to_interconnect_- module_component)	PATH			<pre> interface_mounted_join <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'packaged connector component interface terminal'} shape_aspect => component_interface_terminal </pre>
interface_mounted_join to component_terminal (as access_to_interface_- mounted_component)	PATH			<pre> interface_mounted_join <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'minimally defined component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal')} shape_aspect => component_terminal </pre>
MOVABLE_PACKAGED_COM- PONENT_JOIN_TERMINAL	component_terminal	210		<pre> component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'movable packaged component join terminal'] [shape_aspect.product_definitional = .TRUE.]} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
movable_packaged_component_- join_terminal to length_data_element (as maximum_wire_terminal_length)	PATH			<pre> component_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum wire length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
movable_packaged_component_- join_terminal to length_data_element (as minimum_wire_terminal_length)	PATH			<pre> component_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum wire length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
NEXT_HIGHER_ASSEMBLY_- RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44	213, 214	<pre> (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=] assembly_component_usage) </pre>
reference_designation	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator </pre>
next_higher_assembly_- relationship to data_element (as quantity)	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit </pre>
PACKAGE_TERMINAL_OC- CURRENCE	component_terminal	210		<pre> component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'package terminal occurrence'] [shape_aspect.product_definitional = .TRUE.]} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_terminal_occurrence to package_terminal (as definition)	PATH		333	<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => package_terminal </pre>
package_terminal_occurrence to packaged_component (as associated_component)	PATH		340, 338	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'} product_definition => component_definition => packaged_component </pre>
PACKAGED_COMPONENT_- JOIN_TERMINAL	component_terminal	210	333	<pre> component_terminal <= shape_aspect {shape_aspect [(shape_aspect.description = 'packaged component join terminal')] </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'movable packaged component join terminal')] [shape_aspect.product_definitional = .TRUE.]]
packaged_component_join_- terminal to packaged_component (as associated_component)	PATH		333, 340, 338	component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'} product_definition => component_definition => packaged_component
packaged_component_join_- terminal to packaged_part_join_terminal (as definition)	PATH		333	component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => packaged_part_terminal

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PACKAGED_CONNECTOR_- COMPONENT_INTERFACE_- TERMINAL	component_interface_terminal	210	333	component_interface_terminal <= shape_aspect {[shape_aspect.description = 'packaged connector component interface terminal'] [shape_aspect.product_definitional = .TRUE.]}
packaged_connector_component_- interface_terminal to packaged_connector_component (as associated_component)	PATH		333, 340, 338	component_interface_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {[product_definition.description = 'packaged connector component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']} product_definition => component_definition => packaged_component
packaged_connector_component_- interface_terminal to packaged_part_interface_terminal (as definition)	PATH		333	component_interface_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => packaged_part_terminal </pre>
PCA	assembly_definition	210	338, 268	<pre> assembly_definition <= physical_unit <= product_definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']} </pre>
pca to pca_usage_view (as usage_view)	PATH		338, 268	<pre> assembly_definition <= physical_unit <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'design usage'} product_definition_relationship product_definition_relationship.relatng_- product_definition -> {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (assembly_definition) (externally_defined_physical_unit => externally_defined_assembly_definition) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_assembly_definition) </pre>
PHYSICAL_COMPONENT	component_definition	210		<pre> component_definition <= product_definition {[product_definition product_definition.description != 'laminate component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']} </pre>
physical_component to physical_unit (as usage_definition)	PATH			<pre> component_definition <= product_definition <- product_definition.relationship.related_- product_definition {product_definition.relationship product_definition.relationship.name = 'instantiated part'} product_definition.relationship product_definition.relationship.relating_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design')} </pre>

Table 15 – Mapping table for assembly_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(application_context_element.name = 'physical design usage') product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit)</pre>
ROUTED_PHYSICAL_COMPONENT	routed_physical_component	210		<pre>routed_physical_component <= component_definition <= product_definition {product_definition product_definition.description != 'laminate component'}</pre>
routed_physical_component to curve (as routed_centrelines_shape)	PATH			<pre>routed_physical_component <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition.definition <- property_definition.definition</pre>

Table 15 – Mapping table for assembly_module_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation.items[i] -> representation_item => geometric_representation_item => curve

Table 16 – Mapping table for assembly_module_usage_view UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_MODULE_INTERFACE_TERMINAL	assembly_module_interface_terminal	210		assembly_module_interface_terminal <= assembly_module_terminal <= shape_aspect
ASSEMBLY_MODULE_JOIN_TERMINAL	assembly_module_join_terminal	210		assembly_module_join_terminal <= assembly_module_terminal <= shape_aspect
ASSEMBLY_MODULE_TERMINAL	assembly_module_terminal	210	333, 340, 268	assembly_module_terminal <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition product_definition.name = 'assembly module'}
assembly_module_terminal to assembly_module_usage_view (as associated_definition)	PATH		333, 340, 268	assembly_module_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design_usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) assembly_module_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} assembly_module_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference terminal'}
assembly_module_terminal to connection_zone (as terminal_connection_zone)	PATH		333	
assembly_module_terminal to packaged_part_interface_terminal (as reference_terminal)	PATH			

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => packaged_part_terminal </pre>
assembly_module_terminal to assembly_module_usage_view_- connector_relationship (as related_connector)	PATH			<pre> assembly_module_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.description = 'related connector'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> representation_item => mapped_item => assembly_module_usage_view_connector_- relationship </pre>
ASSEMBLY_MODULE_USAGE_VIEW	(physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)	210 210 210	268	<pre> (physical_unit <= product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition)) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']} </pre>
assembly_module_usage_view to functional_unit_usage_view (as implemented_function)	PATH		338	<pre> (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'implemented function'} product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage' } </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
assembly_module_usage_view to datum_based_length_measure (as maximum_negative_- component_height)	PATH		338	<pre> (product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'maximum negative component height'} </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_module_usage_view to datum_based_length_measure (as maximum_positive_-component_height)	PATH			(physical_unit <= product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) product_definition characterized_product_definition = product_-definition characterized_product_definition characterized_definition = characterized_-product_definition characterized_definition <- property_definition.definition property_definition.definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_-representation -> representation {representation.name = 'maximum positive component height'}
ASSEMBLY_MODULE_USAGE_VIEW_CONNECTOR_RELATIONSHIP	assembly_module_usage_view_-connector_relationship	210	16	assembly_module_usage_view_connector_-relationship <= mapped_item {mapped_item mapped_item.mapping_source -> representation_map representation_map.mapping_origin ->

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => geometric_representation_item => placement} </pre>
externally_visible_partial_reference_designation	PATH			<pre> assembly_module_usage_view_connector_- relationship <= mapped_item <= representation_item <- representation.items[i] representation <- representation_relationship.rep_2 {representation_relationship representation_relationship.name = 'reference designation_assignment'} representation_relationship.rep_1 -> representation {representation representation.name = 'externally visible partial reference designation'} representation.items[i] -> representation_item => descriptive_representation_item {descriptive_representation_item descriptive_representation_item.name = 'partial reference designation'} descriptive_representation_item.description </pre>
assembly_module_usage_view_-connector_relationship to packaged_connector (as associated_usage)	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_- representation </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => (physical_unit => packaged_part => packaged_connector) (physical_unit => externally_defined_physical_unit => externally_defined_packaged_part => externally_defined_packaged_connector) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector) mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> </pre>
assembly_module_usage_view - connector relationship to assembly_module_usage_view (as associating_usage)	PATH			<pre> representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {[product_definition.name = 'assembly module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
PACKAGED_CONNECTOR	<pre> (packaged_connector) (externally_defined_packaged_- connector) (library_defined_packaged_- connector) </pre>	<pre> 210 210 210 </pre>	338	<pre> (packaged_connector <= packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_connector <= externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_connector <= library_defined_packaged_part <= </pre>

Table 16 – Mapping table for assembly_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition])
PcA_TERMINAL	assembly_module_terminal	210	333	assembly_module_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'pca terminal'}
pca_terminal to pca_usage_view (as associated_definition)	PATH		333, 340, 268, 338	assembly_module_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => physical_unit => (assembly_definition) (externally_defined_physical_unit => externally_defined_assembly_definition) (externally_defined_physical_unit =>

Table 16 – Mapping table for assembly_module_usage_view UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				library_defined_physical_unit => library_defined_assembly_definition)
PCA_USAGE_VIEW	(assembly_definition) (externally_defined_assembly_ - definition) (library_defined_assembly_ - definition)	210 210 210	268	(assembly_definition <= physical_unit <= product_definition) (externally_defined_assembly_definition <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_assembly_definition <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']}

Table 17 – Mapping table for assembly_physical_interface_requirement UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_PLACEMENT_- RESTRICTION_ASSIGNMENT	representation	43		representation {representation.name = 'component placement restriction assignment'}
components_permitted	descriptive_representation_item	45	109	representation representation.items[i] -> {representation_item representation_item.name = 'components permitted'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}}
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation representation {representation.name = 'maximum mounting clearance'}
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_negative_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation representation {representation.name = 'maximum negative component height'}

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_positive_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'maximum positive component height'}
component_placement_- restriction_assignment to datum_based_length_measure (as minimum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'minimum mounting clearance'}
component_placement_- restriction_assignment to item_restricted_requirement_- occurrence (as requirement)	PATH		42, 340	representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition => requirements_property => grouped_requirements_property {[grouped_requirements_property <= group group.name = 'item restricted requirements property']}
component_placement_- restriction_assignment to mounting_restriction_area (as area)	PATH		42, 340, 333	representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement restriction assignment area'} representation_relationship.rep_2 ->

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_area </pre>
component_placement_- restriction_assignment to mounting_restriction_volume (as volume)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement_restriction_assignment volume'} representation_relationship.rep_2 -> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_volume </pre>

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERFACE_REQUIREMENT_- OCCURRENCE	requirements_property	210	340	requirements_property <= property_definition {property_definition property_definition.description = 'interface requirement'}}
interface_requirement_occurrence to next_higher_assembly_- interface_requirement (as interface_to_next_higher_- assembly)	PATH		340, 265, 266, 338, 339, 354	requirements_property <= property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'interface to next higher assembly'} property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement'} product_definition <- product_definition_relationship.related_- product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'higher assembly interface'}}

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship => product_definition_usage => assembly_component_usage => specified_higher_usage_occurrence </pre>
MATING_CONNECTOR_COMPONENT	(component_definition) ([component_definition] [product_definition_with_ - associated_documents])	210 210 41	338	<pre> (component_definition <= product_definition {product_definition product_definition.description = 'mating connector'}) ([product_definition_with_associated_ documents] [component_definition <= product_definition {product_definition product_definition.description = 'mating connector'}]) </pre>
reference_designation	product_definition.id	41		<pre> component_definition <= product_definition product_definition.id </pre>
mating_connector_component to ee_specification (as definition)	PATH		338	<pre> component_definition <= product_definition => product_definition_with_associated_documents product_definition_with_associated_ documents.documentation_ids[i] -> document => ee_specification </pre>
mating_connector_component to packaged_connector (as definition)	PATH		338	<pre> component_definition <= product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship </pre>

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship.relatiing_ product_definition -> product_definition => physical_unit => (packaged_part => packaged_connector) (externally_defined_physical_unit => externally_defined_packaged_part => externally_defined_packaged_connector) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector) </pre>
MATING_CONNECTOR_- PLACEMENT_RELATIONSHIP	shape_representation	41	111, 208, 342	<pre> {shape_representation <= representation representation.name = 'mating connector placement'} </pre>
mating_connector_placement_- relationship to axis_placement (as connector_placement)	PATH		342, 109	<pre> shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'connector placement'} representation_item => geometric_representation_item => placement </pre>
mating_connector_placement_- relationship to mating_connector_component (as placed_connector)	PATH		265, 266, 338	<pre> shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.description = 'mating connector'} product_definition => component_definition </pre>
<pre> mating_connector_placement_- relationship to next_higher_assembly_interface_- requirement (as interface_context) </pre>	PATH		<pre> 342, 337, 339, 265, 266, 338, 354 </pre>	<pre> shape_representation <= representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'volume shape'] [representation => shape_representation]} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition </pre>

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition_relationship {product_definition_relationship product_definition_relationship.relatin product_definition -> [product_definition product_definition.description = 'mating connector'] [product_definition => component_definition]} product_definition_relationship => product_definition_usage => assembly_component_usage => specified_higher_usage_occurrence
NEXT_HIGHER_ASSEMBLY_ INTERFACE_REQUIREMENT	specified_higher_usage_ occurrence	44	212	
next_higher_assembly_interface_ requirement to bound_volume_shape (as shape)	PATH		354, 342	specified_higher_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation ->

Table 17 – Mapping table for assembly_physical_interface_requirement UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				{representation representation.name = '3d bound volume shape'} representation => shape_representation
next_higher_assembly_interface - requirement to ee_product_version (as version_of_next_higher_ assembly)	PATH		265, 266, 263, 264	specified_higher_usage_occurrence specified_higher_usage_occurrence.upper_usage -> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition product_definition.formation -> product_definition_formation

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_REQUIREMENT_- ALLOCATION	(product_definition_relationship) (shape_aspect_relationship)	41 41		{product_definition_relationship product_definition_relationship.name = 'assembly_requirement_allocation'} {shape_aspect_relationship shape_aspect_relationship.name = 'assembly requirement_allocation'}
assembly_requirement_allocation to assembly_component (as assembly_assignment)	PATH		17, 338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition => component_definition)
assembly_requirement_allocation to assembly_joint (as assembly_assignment)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint
assembly_requirement_allocation to component_feature (as assembly_assignment)	PATH		17, 333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature)
assembly_requirement_allocation to ee_requirement_occurrence (as requirement)	PATH		17	(product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition product_definition.frame_of_reference ->

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_context <= application_context_element (application_context_element.name = 'design requirement') (application_context_element.name = 'requirement') characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => requirements_property (shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition => requirements_property) </pre>
ASSEMBLY_SEPARATION_REQUIREMENT_ALLOCATION	(product_definition_relationship) (shape_aspect_relationship)	41 41		<pre> ({product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'assembly_separation_requirement']}) ({shape_aspect_relationship [shape_aspect_relationship.name = 'assembly requirement_allocation'] [shape_aspect_relationship.description = 'assembly_separation_requirement']}) </pre>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_separation_- requirement_allocation to assembly_component (as disjoint_assignment)	PATH		17, 338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition => component_definition)
assembly_separation_- requirement_allocation to assembly_joint (as disjoint_assignment)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint
assembly_separation_- requirement_allocation to component_feature (as disjoint_assignment)	PATH		17, 18, 333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature)
ELECTROMAGNETIC_COM- PATIBILITY_REQUIREMENT_- ALLOCATION	product_definition_relationship	41		{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'electromagnetic compatibility requirement allocation']}]
INTERFACE_COMPONENT_- ALLOCATION	product_definition_relationship	41	195	{product_definition_relationship product_definition_relationship.name = 'interface_component_allocation'}
interface_component_allocation to interface_component (as mating_solution)	PATH		265, 266	product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition [product_definition =>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> component_definition => interface_component_definition] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'']] {product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.related_- product_definition -> [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'] [product_definition => (physical_unit) externally_defined_physical_unit => library_defined_physical_unit]]} product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.description = 'mating connector'} product_definition => component_definition </pre>
interface_component_allocation to mating_connector_component (as mating_requirement)	PATH			<pre> product_definition_relationship product_definition_relationship.related_- product_definition product_definition.description = 'mating connector'} product_definition => component_definition </pre>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
REQUIREMENT_ALLOCATION	(product_definition_relationship) (shape_aspect_relationship) (requirement_allocation_group)	41 41 210		<pre> ({product_definition_relationship product_definition_relationship.name = 'requirement_allocation'}) ({shape_aspect_relationship shape_aspect_relationship.name = 'requirement_allocation'}) ({{requirement_allocation_group <= property_definition_relationship} [requirement_allocation_group <= group]} requirement_allocation_group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = product) (group_assigned_item = product_definition_ formation) (group_assigned_item = configuration_item)}) </pre>
requirement_allocation to component_external_reference (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = representation {representation.name = 'component external reference'} </pre>
requirement_allocation to component_feature_external_ reference (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => </pre>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment group_assigned_item = representation_item {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item </pre>
requirement_allocation to component_feature (as target_product_object)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminate_component_interface_terminal) (component_feature) </pre>
requirement_allocation to ee_product (as target_product_object)	PATH		268	<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product </pre>
requirement_allocation to ee_product_configuration (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = configuration_item </pre>
requirement_allocation to ee_product_definition (as target_product_object)	PATH			<pre> product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => (product_definition_with_associated_documents) (physical_unit) </pre>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
requirement_allocation to ee_product_version (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product_definition_ formation </pre>
requirement_allocation to ee_requirement_occurrence (as allocated_requirement)	PATH			<pre> (product_definition_relationship product_definition_relationship.relation_ product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition </pre>

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition => (shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => (requirement_allocation_group <= property_definition_relationship property_definition_relationship.relatng_- property_definition ->

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition => requirements_property
requirement_allocation to ee_requirement_occurrence (as target_product_object)	PATH			requirement_allocation_group <= property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'derived from operation'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property
requirement_allocation to assembly_component (as target_product_object)	PATH		333	product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition => component_definition)
requirement_allocation to part_feature (as target_product_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface')

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
requirement_allocation to physical_connectivity_definition (as target_product_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
requirement_allocation to physical_connectivity_element (as target_product_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_element
SHIELD_ALLOCATION	product_definition_relationship	41	17, 309	{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'shield_allocation']}
shield_allocation to electromagnetic_requirement_- occurrence (as requirement)	PATH		17	product_definition_relationship product_definition_relationship.relationing_- product_definition -> {product_definition <- product_definition_relationship.relationing_- product_definition product_definition_relationship product_definition_relationship.name = 'shield group member'} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'electromagnetic requirement'} property_definition => requirements_property product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition (product_definition =>
shield_allocation to shield (as assembly_assignment)	PATH		340, 333	

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				component_definition => discrete_shield)
THERMAL_REQUIREMENT_- ALLOCATION	product_definition_relationship	41	360	{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'thermal_requirement_allocation']}]
thermal_requirement_allocation to thermal_component (as assembly_assignment)	PATH		17, 338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.description = 'thermal component'} (product_definition => component_definition => thermal_component)
thermal_requirement_allocation to thermal_requirement_occurrence (as requirement)	PATH		17	product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement'} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'thermal requirement'}

Table 18 – Mapping table for assembly_physical_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition => requirements_property

Table 19 – Mapping table for assembly_shape UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_COMPONENT_- 2D_SHAPE	shape_representation	41	111, 340, 338, 333	<pre> {shape_representation <= [representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition)]} </pre>
assembly_component_2d_shape to assembly_component (as shape_characterized_component)	PATH		340, 338, 333	<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation.definition -> </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition)</pre>
ASSEMBLY_COMPONENT_- 3D_SHAPE	(shape_representation)	41	111, 340, 338, 333	<pre>(shape_representation <= representation {representation representation.name = '3d bound volume shape'})</pre>
assembly_component_3d_shape to assembly_component (as shape_characterized_component)	PATH		340, 338, 333	<pre>(shape_representation <= representation <-) property_definition.used_ representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition = product_ definition)</pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => component_definition)
BOND_ASSEMBLY_POSITION	mapped_item	43		{mapped_item <= representation_item representation_item.name = 'assembly position'}
bond_assembly_position to assembled_with_bonding (as bond)	PATH		109, 344	mapped_item <= representation_item <- representation.items[i] {representation representation.name = 'bond assembly position'} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'assembled with bonding'} shape_aspect_relationship => assembly_joint mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
bond_assembly_position to axis_placement_3d (as bond_definition_placement)	PATH			

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
bond_assembly_position to default_assembly_bond_shape (as bond_model)	PATH		339, 354, 340	mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition [property_definition.description = 'default assembly bond shape'] [property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage[]} property_definition => product_definition_shape mapped_item <= representation_item <- representation.items[i] {representation [representation.name = '3d bound volume shape'] representation <-
bond_assembly_position to physical_unit_3d_shape (as assembly_model)	PATH		109, 340, 338, 111	

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition = product_ - definition product_definition => physical_unit[]} representation => shape_representation
COMPONENT_2D_EDGE_LO- CATION	component_location	210	342, 343	component_location <= representation {representation representation.context_of_items -> [representation.context => geometric_representation_context space_dimension = 2] [representation_context representation_context.context_type = 'component edge']}
component_2d_edge_location to assembly_joint (as mounting_surface_assembly_- joint)	PATH			component_location <= representation representation < - property_definition_representation.used_ - representation property_definition_representation {property_definition_representation

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.name = 'mounting surface assembly joint' property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_joint </pre>
component_2d_edge_location to assembly_joint (as reference_terminal_assembly_- joint)	PATH			<pre> component_location <= representation representation <- property_definition_representation.used_- representation property_definition_representation {property_definition_representation property_definition_representation.name = 'reference terminal assembly joint'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_joint </pre>
COMPONENT_2D_LOCATION	component_location	210		<pre> component_location <= representation ({representation </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation.context_of_items -> representation_context {representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 2}}
placement_fixed	descriptive_representation_- item.description	45		component_location <= representation representation.items[i] -> {representation_item representation_item.name = 'placement fixed'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}}
component_2d_location to cartesian_transformation_- operator_2d (as transformation)	PATH		342	component_location <= representation representation.items[i] -> representation_item => geometric_representation_item => cartesian_transformation_operator => cartesian_transformation_operator_2d
COMPONENT_2D_STACKED_- LOCATION	component_location	210		component_location <= representation {representation representation.context_of_items -> [representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 2]}

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_2d_stacked_location to assembly_joint (as mounting_joint)	PATH			[representation_context representation_context.context_type = 'component_stacked']] component_location <= representation <= property_definition_representation.used_ representation property_definition_representation {property_definition_representation property_definition_representation.name = 'mounting_joint'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_joint
COMPONENT_2D_SURFACE_ LOCATION	component_location	210		component_location <= representation {representation.context_of_items -> representation.context => [representation_context geometric_representation_context geometric_representation_context.coordinate_ space_dimension = 2] [representation_context representation_context.context_type = 'component_surface']}]
component_2d_surface_location to assembly_joint (as mounting_surface_assembly_ joint)	PATH			component_location <= representation representation <= property_definition_representation.used_ representation property_definition_representation {property_definition_representation

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.name = 'mounting surface assembly joint' property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_joint </pre>
COMPONENT_3D_LOCATION	component_location	210	342, 343	<pre> component_location <= representation {representation representation.context_of_items -> representation_context => geometric_representation_context space_dimension = 3} </pre>
placement_fixed	descriptive_representation_item	45	342	<pre> component_location <= representation representation.items[i] -> {representation_item representation_item.name = 'placement fixed'} descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true')} (descriptive_representation_item.description = 'false')} </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_3d_location to axis_placement_3d (as transformation)	PATH			<pre> component_location <= representation representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>
COMPONENT_ASSEMBLY_- 2D_POSITION	mapped_item	43	308	<pre> {mapped_item <= representation_item representation_item.name = 'component assembly 2d position'} </pre>
component_assembly_2d_position to assembly_component_2d_shape (as component_model)	PATH		340, 338, 333, 111	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin']} [representation_item => geometric_representation_item => axis2_placement_2d]} representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape']} [representation representation.items[i] -> representation_item => geometric_representation_item => axis2_placement_2d] [representation <- property_definition_representation.used_ - property_definition_representation property_definition_representation.definition -> </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition)]} representation => shape_representation </pre>
component_assembly_2d_position to component_2d_location (as transformation)	PATH		109, 342	<pre> mapped_item <= {mapped_item.mapping_target -> representation_item representation_item => geometric_representation_item => cartesian_transformation_operator => cartesian_transformation_operator_2d} representation_item <- representation.items[i] representation representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 2} representation => component_location} representation_item <- representation.items[i] </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation representation.context_of_items -> representation.context => geometric_representation_context.coordinate_- space_dimension = 2} representation => component_location mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage]) </pre>
component_assembly_2d_position to next_higher_assembly_- relationship (as component)	PATH		340, 339, 354	
component_assembly_2d_position to physical_unit_planar_shape (as assembly_model)	PATH		338, 111	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape']} [representation <- </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition => physical_unit[] representation => shape_representation
COMPONENT_ASSEMBLY_- 3D_POSITION	mapped_item	43	308	{mapped_item <= representation_item representation_item.name = 'component assembly 3d position'}
component_assembly_3d_position to assembly_component_3d_shape (as component_model)	PATH		340, 338, 333, 111	mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => placement => axis2_placement_3d[]} representation_map.mapped_representation -> {[representation (representation.name = '3d bound volume shape')]

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.name = 'part template non planar 2d shape') representation representation.items[i] -> representation_item representation_item.name = 'origin' representation representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition)]} representation => shape_representation mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => </pre>
component_assembly_3d_position to component_3d_location (as transformation)	PATH		342	

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> placement => axis2_placement_3d{ representation_item <- representation.items[i] representation => component_location </pre>
component_assembly_3d_position to next_higher_assembly_- relationship (as component)	PATH		340, 339, 354	<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage]) </pre>
component_assembly_3d_position to physical_unit_3d_shape (as assembly_model)	PATH		338, 111	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape']} [representation </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit]] representation => shape_representation </pre>
COMPONENT_PART_2D_GEO- METRIC_REPRESENTATION_- RELATIONSHIP	shape_representation_relationship	41	40	<pre> {shape_representation_relationship <= representation_relationship {representation_relationship representation_relationship.name = 'component part planar shape'}} </pre>
component_part_2d_geometric_- representation_relationship to assembly_component_2d_shape (as component_shape)	PATH		340, 338, 333, 111	<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation [representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition)]} representation => shape_representation </pre>
component_part_2d_geometric_- representation_relationship to part_template_planar_shape (as part_shape)	PATH		340	<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
component_part_2d_geometric_- representation_relationship to physical_unit_planar_shape (as part_shape)	PATH		338	<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit[]} representation => shape_representation
COMPONENT_PART_3D_GEO- METRIC_REPRESENTATION_- RELATIONSHIP	shape_representation_relationship	41		shape_representation_relationship <= representation_relationship {representation_relationship representation_relationship.name = 'component part 3d shape'}
component_part_3d_geometric_- representation_relationship to assembly_component_3d_shape (as component_shape)	PATH		41, 338, 333, 111	shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation [(representation.name = '3d bound volume shape') (representation.name = 'part template non planar 2d shape')]} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition)

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> }} representation => shape_representation shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation </pre>
component_part_3d_geometric_- representation_relationship to part_template_3d_shape (as part_shape)	PATH		333, 111, 41	
component_part_3d_geometric_- representation_relationship to part_template_non_planar_2d_- shape (as part_shape)	PATH			

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
component_part_3d_geometric_ representation_relationship to physical_unit_3d_shape (as part_shape)	PATH		338, 111, 41	shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition =>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit[] representation => shape_representation
DESIGN_VIEW_CONNECTION_ZONE_POSITION_RELATIONSHIP	representation_relationship_with_- transformation	43		{representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'design view connection zone position relationship'}
design_view_connection_zone_- terminal_shape_relationship to shape_definition (as associating_design_view_shape)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'design view shape'} representation => shape_representation
design_view_connection_zone_- terminal_shape_relationship to connection_zone (as associated_usage)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => shape_representation} representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
design_view_connection_zone_- position_relationship to connection_zone (as associated_usage)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => shape_representation} representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
design_view_connection_zone_- position_relationship to shape_definition (as associated_connection_zone_- shape_definition)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} representation => shape_representation

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_view_connection_zone_- position_relationship to axis_placement (as associated_usage_placement)	PATH			<pre> representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement </pre>
design_view_connection_zone_- position_relationship to physical_unit_design_view (as associating_design_view)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> representation {representation.name = 'position'} {representation => shape_representation} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit}</pre>
DESIGN_VIEW_TERMINAL_- COMPONENT_SHAPE_RELATIONSHIP	representation_relationship_with_- transformation	43		<pre>{representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'design view terminal component shape relationship'}</pre>
design_view_terminal_- component_shape_relation feature_shape_definition (as associating_component_shape)	PATH			<pre>representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation (representation.name = 'planar projected shape') (representation.name = '3d bound volume shape')} representation => shape_representation}</pre>
design_view_terminal_- component_shape_relationship to conductive_interconnect_- element_terminal (as associated_usage)	PATH			<pre>representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation (representation.name = 'terminal shape')} representation => shape_representation}</pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} representation_relationship_with_ transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} representation => shape_representation representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item </pre>
design_view_terminal_- component_shape_relationship to feature_shape_definition (as associated_terminal_shape_- definition)	PATH			
design_view_terminal_- component_shape_relationship to axis_placement (as associated_usage_placement)	PATH			

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement </pre>
design_view_terminal_- component_shape_relationship to interconnect_component_- interface_terminal (as associated_usage)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} {representation => shape_representation} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_interface_terminal </pre>
design_view_terminal_- component_shape_relationship to interconnect_component_join_- terminal (as associated_usage)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} {representation => shape_representation} </pre>

Table 19 – Mapping table for assembly_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_terminal </pre>
<pre> design_view_terminal_- component_shape_relationship_to movable_packaged_component_- join_terminal (as associated_usage) </pre>	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} {representation => shape_representation} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect </pre>

Table 19 – Mapping table for assembly_shape UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => {shape_aspect shape_aspect.description = 'movable packaged component join terminal'} component_terminal

Table 20 – Mapping table for assembly_technology UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_BOND_DEFINITION	assembly_bond_definition	210	333	<pre> assembly_bond_definition <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition.name = 'assembly bond model'} </pre>
name	shape_aspect.name	41	333	<pre> assembly_bond_definition <= shape_aspect shape_aspect.name </pre>
assembly_bond_definition to analytical_representation (as analytical_assembly_bond_- representation)	PATH			<pre> assembly_bond_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation => analytical_representation </pre>
assembly_bond_definition to externally_defined_ee_product_- definition (as associated_definition)	PATH			<pre> assembly_bond_definition <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => externally_defined_product_definition </pre>
assembly_bond_definition to ee_material (as bond_material)	PATH		333	<pre> assembly_bond_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition </pre>

Table 20 – Mapping table for assembly_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- material_designation.definitions[i] material_designation </pre>
assembly_bond_definition to external_definition (as bond_category)	PATH			<pre> assembly_bond_definition group_assigned_item = assembly_bond_definition group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => bond_category </pre>
assembly_bond_definition to part_feature (as bonded_feature_1)	PATH			<pre> assembly_bond_definition <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'bonded feature 1'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') </pre>

Table 20 – Mapping table for assembly_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)} assembly_bond_definition <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'bonded feature 2'} </pre>
assembly_bond_definition to part_feature (as bonded_feature_2)	PATH			

Table 20 – Mapping table for assembly_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => </pre>

Table 20 – Mapping table for assembly_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
ASSEMBLY_TECHNOLOGY_- SPECIFICATION	ee_specification	210		<pre> ee_specification <= document {document {document.kind -> document_type document_type.product_data_type = 'assembly technology_specification'}} </pre>
DEFAULT_ASSEMBLY_- BOND_SHAPE	shape_representation	41	94	<pre> shape_representation <= representation {representation.name = 'default assembly bond shape'} </pre>
default_assembly_bond_shape to assembly_bond_definition (as characterized_bond_definition)	PATH		333	<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 20 – Mapping table for assembly_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_bond_definition
REFERENCE_PACKAGED_- PART_ASSEMBLY_IMPL- EMENTATION	reference_packaged_part_- assembly_implementation	210		reference_packaged_part_assembly_- implementation <= physical_unit <= product_definition {[product_definition product_definition.formation -> product_definition.formation product_definition.formation.of_product -> product <= product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition product_definition.name = 'assembly module'] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout design usage']]
REFERENCE_PACKAGED_- PART_INTERCONNECT_IM- PLEMENTATION	reference_packaged_part_- interconnect_implementation	210		reference_packaged_part_interconnect_- implementation <= physical_unit <= product_definition {[product_definition product_definition.formation -> product_definition.formation

Table 20 – Mapping table for assembly_technology UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model' [product_definition product_definition.name = 'interconnect module'] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']]}</pre>

Table 21 – Mapping table for authorization UoF

Application element	AIM element	Source	Rules	Reference path
CONTRACT	contract	41	100, 67	
contract_number	contract.name	41		
contract to date (as contract_date)	PATH		102, 289, 101, 350	<pre> contract date_assigned_item = contract date_assigned_item <- applied_date_assignment.items[i] {date_assignment {date_assignment.role -> date_role date_role.name = 'contract date'} date_assignment date_assignment.assigned_date -> date => calendar_date </pre>
contract to date_and_time (as contract_date)	PATH		103	<pre> contract date_and_time_assigned_item = contract date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] {date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'contract date'} date_and_time_assignment date_and_time_assignment.assigned_date_and_ time -> date_and_time </pre>

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
contract to ee_approval (as contract_approval)	PATH		13	contract approval_assigned_item = contract approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
contract to organization (as contractor)	PATH		106	contract organization_assigned_item = contract organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'contractor'} organization_assignment organization_assignment.assigned_organization -> organization
contract to person_and_organization (as contractor)	PATH		108	contract person_and_organization_assigned_item = contract person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'contractor'} person_and_organization_assignment person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATE	calendar_date	41	73	
day	calendar_date.day_component	41		
month	calendar_date.month_component	41		
year	date.year_component	41	350	calendar_date <= date date.year_component
DATE_AND_TIME	date_and_time	41		
year	PATH			date_and_time.date_component -> date => calendar_date calendar_date.year_component
month	PATH			date_and_time.date_component -> date => calendar_date calendar_date.month_component
day	PATH			date_and_time.date_component -> date => calendar_date calendar_date.day_component
hour	local_time.hour_component	41		date_and_time date_and_time.time_component -> local_time local_time.hour_component
minute	local_time.minute_component	41		date_and_time date_and_time.time_component -> local_time local_time.minute_component
second	local_time.second_component	41	72	date_and_time date_and_time.time_component -> local_time local_time.second_component
time_zone	coordinated_universal_time_- offset.hour_offset	41		date_and_time date_and_time.time_component -> local_time

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				local_time.zone -> coordinated_universal_time_offset coordinated_universal_time_offset.hour_offset
time_zone_sense	coordinated_universal_time_- offset.sense	41		date_and_time date_and_time.time_component -> local_time local_time.zone -> coordinated_universal_time_offset (coordinated_universal_time_offset.sense = .ahead.) (coordinated_universal_time_offset.sense = .behind.)
EE_APPROVAL	approval	41		
purpose	approval.level	41		
approval_status	approval_status	41	98, 288	approval approval_status -> approval_status {approval_status (approval_status.name = 'approved') (approval_status.name = 'not yet approved') (approval_status.name = 'disapproved') (approval_status.name = 'withdrawn')}}
ee_approval to date (as signoff_date)	PATH		350	approval <- approval_date_time.dated_approval approval_date_time approval_date_time.date_time -> date_time_select date_time_select = date date => calendar_date
ee_approval to date_and_time (as signoff_date)	PATH		10	approval <- approval_date_time.dated_approval approval_date_time approval_date_time.date_time -> date_time_select

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_approval to organization (as authorizer)	PATH			<pre> date_time_select = date_and_time date_and_time approval <- approval_person_organization.authorized_- approval approval_person_organization {approval_person_organization approval_person_organization.role -> approval_role approval_role.role = 'authorizer'} approval_person_organization.person_- organization -> person_organization_select person_organization_select = organization organization </pre>
ee_approval to person_and_organization (as authorizer)	PATH		11	<pre> approval <- approval_person_organization.authorized_- approval approval_person_organization {approval_person_organization approval_person_organization.role -> approval_role.role = 'authorizer'} approval_person_organization.person_- organization -> person_organization_select person_organization_select = person_and_- organization person_and_organization </pre>
ORGANIZATION	organization	41	219	
address	organizational_address	41	346	<pre> organization <- organizational_address.organizations[i] organizational_address </pre>
identifier	organization.id	41		

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PERSON	person	41		
address	personal_address	41	346	person <- personal_address.people[i] personal_address
identifier	person.id	41		
PERSON_AND_ORGANIZATION	person_and_organization	41	238	
person_id_in_organization	person_and_organization.name	41		
person_and_organization to organization (as assigned_organization)	PATH			person_and_organization person_and_organization.the_organization -> organization
person_and_organization to person (as assigned_person)	PATH			person_and_organization person_and_organization.the_person -> person
SECURITY_CLASSIFICATION	security_classification	41	110	
classification_level	security_classification_level.name	41		security_classification.security_level -> security_classification.name
security_classification to date (as classification_date)	PATH		296, 102, 289, 101, 350	security_classification date_assigned_item = security_classification date_assigned_item <- applied_date_assignment.items[i] applied_date_assignment <= {date_assignment date_assignment.role -> date_role date_role.name = 'classification date'} date_assignment date_assignment.assigned_date -> date => calendar_date

Table 21 – Mapping table for authorization UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
security_classification to date_and_time (as classification_date)	PATH		72	security_classification date_and_time_assigned_item = security_ classification date_and_time_assigned_item <= applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'classification date'} date_and_time_assignment date_and_time_assignment.assigned_date_and_ time -> date_and_time
security_classification to date_and_time (as declassification_date)	PATH		296, 103	security_classification date_and_time_assigned_item = security_ classification date_and_time_assigned_item <= applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'declassification date'} date_and_time_assignment date_and_time_assignment.assigned_date_and_ time -> date_and_time
security_classification to ee_approval (as security_classification_approval)	PATH		13	security_classification approval_assigned_item = security_ classification approval_assigned_item <= applied_approval_assignment.items[i]

Table 21 – Mapping table for authorization UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<code>applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval</code>
security_classification to person_organization (as classification_officer)	PATH			<code>security_classification person_and_organization_assigned_item = security_classification person_and_organization_assigned_item <- applied_person_and_organization_assignment.items[i] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'classification_officer'} person_and_organization_assignment person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization</code>

Table 22 – Mapping table for bare_die UoF

Application element	AIM element	Source	Rules	Reference path
BARE_DIE	(bare_die) (externally_defined_bare_die) (library_defined_bare_die)	210 210 210	338, 19	(bare_die <= physical_unit <= product_definition) (externally_defined_bare_die <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_bare_die <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.formation -> product_definition.formation product_definition.formation.of_product -> product <= product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'bare die'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']}

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
bare_die to functional_unit_usage_view (as implemented_function)	PATH		338	<pre> (bare_die <=) (externally_defined_bare_die <= externally_defined_physical_unit <=) (library_defined_bare_die <= library_defined_physical_unit <=) (externally_defined_physical_unit <=) physical_unit <= product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'implemented_function'} product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional design_usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
bare_die to cartesian_point (as least_material_condition_- centroid_location)	PATH			<pre> (bare_die <=) (externally_defined_bare_die <= externally_defined_physical_unit <=) (library_defined_bare_die <= library_defined_physical_unit <=) </pre>

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'material property data'} representation representation.items[i] -> representation_item => {cartesian_point cartesian_point.name = 'least material condition centroid location'} cartesian_point (bare_die <= (externally_defined_bare_die <= externally_defined_physical_unit <= (library_defined_bare_die <= library_defined_physical_unit <=
bare_die to cartesian_point (as maximum_material_ condition_centroid_location)	PATH			(bare_die <= (externally_defined_bare_die <= externally_defined_physical_unit <= (library_defined_bare_die <= library_defined_physical_unit <=

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'material property data'} representation representation.items[i] -> representation_item => {cartesian_point cartesian_point.name = 'maximum material condition centroid location'} cartesian_point (bare_die <= externally_defined_bare_die <= externally_defined_physical_unit <= (library_defined_bare_die <= library_defined_physical_unit <= externally_defined_physical_unit <=)
bare_die to length_data_element (as maximum_height_above_ seating_plane)	PATH			

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> physical_unit <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation.name = 'mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum height above seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (bare_die <=) (externally_defined_bare_die <= externally_defined_physical_unit <=) (library_defined_bare_die <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_- definition </pre>
bare_die to technology (as unit_technology)	PATH			

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'device unit technology'} property_definition_relationship.relatin_ property_definition -> property_definition {property_definition property_definition.name = 'unit technology'} property_definition
BARE_DIE_BOTTOM_SUR- FACE	bare_die_bottom_surface	210		bare_die_bottom_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect
bare_die_bottom_surface to bare_die (as associated_definition)	PATH			bare_die_bottom_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition {{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage'}} product_definition => (physical_unit => bare_die) (externally_defined_product_definition => externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_product_definition => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)}} </pre>
BARE_DIE_EDGE_SEGMENT_- SURFACE	bare_die_edge_segment_surface	210		<pre> [bare_die_edge_segment_surface <= shape_aspect] [bare_die_edge_segment_surface <= shape_aspect_relationship] </pre>
bare_die_edge_segment_surface to bare_die (as associated_definition)	PATH			<pre> bare_die_edge_segment_surface <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition {{product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element design usage'}} product_definition => (physical_unit => bare_die) (externally_defined_product_definition => externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_product_definition => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)}} </pre>
bare_die_edge_segment_surface to bare_die_edge_surface (as composed_surface)	PATH			<pre> bare_die_edge_segment_surface <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => </pre>

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				part_mounting_feature => bare_die_surface => bare_die_edge_surface
bare_die_edge_segment_surface to edge_segment_vertex (as end_vertex)	PATH			bare_die_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
bare_die_edge_segment_surface to edge_segment_vertex (as start_vertex)	PATH			bare_die_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
BARE_DIE_EDGE_SURFACE	bare_die_edge_surface	210		bare_die_edge_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect
bare_die_edge_surface to bare_die (as associated_definition)	PATH			bare_die_edge_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect shape_aspect.of_shape ->

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition {{product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'}} product_definition => (physical_unit => bare_die) (externally_defined_product_definition => externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_product_definition => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)}
BARE_DIE_SURFACE	(bare_die_surface) (bare_die_edge_segment_surface)	210 210		(bare_die_surface <= part_mounting_feature <= shape_aspect) (bare_die_edge_segment_surface <= shape_aspect)

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
BARE_DIE_TERMINAL	bare_die_terminal	210	333	bare_die_terminal <= minimally_defined_bare_die_terminal <= shape_aspect
bare_die_terminal to connection_zone (as external_connection_area)	PATH		333	bare_die_terminal <= minimally_defined_bare_die_terminal <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'external connection area'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
bare_die_terminal to connection_- zone_interface_plane_relationship (as seating_plane_zone)	PATH			bare_die_terminal <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'seating plane zone'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => connection_zone_interface_plane_relationship
bare_die_terminal to ee_material (as material)	PATH		333	bare_die_terminal <= minimally_defined_bare_die_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= material_designation.definitions[i] material_designation

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
BARE_DIE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'bare die terminal surface constituent relationship'}
bare_die_terminal_surface_- constituent_relationship to bare_die_terminal (as related_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal
bare_die_terminal_surface_- constituent_relationship to bare_die_surface (as relating_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_mounting_feature => bare_die_surface
BARE_DIE_TOP_SURFACE	bare_die_top_surface	210		bare_die_top_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect
bare_die_top_surface to bare_die (as associated_definition)	PATH			bare_die_top_surface <= bare_die_surface <= part_mounting_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {product_definition product_definition.frame_of_reference ->

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_context <= application_context_element application_context_element.name = 'physical design usage' product_definition => (physical_unit => bare_die) (externally_defined_product_definition => externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_product_definition => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die) </pre>
CONNECTION_ZONE_INTERFACE_PLANE_RELATIONSHIP	connection_zone_interface_- plane_relationship	210		<pre> {[connection_zone_interface_plane_relationship <= shape_aspect_relationship] [connection_zone_interface_plane_relationship <= shape_aspect]} </pre>
derived_zone_shape_class	shape_aspect_- relationship.description			<pre> connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.description = {(shape_aspect_relationship.description = 'area') (shape_aspect_relationship.description = 'edge')} </pre>
connection_zone_interface_- plane_relationship to connection_zone (as derived_zone)	PATH			<pre> connection_zone_interface_plane_relationship <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived zone'} </pre>

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} connection_zone_interface.plane_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} connection_zone_interface.plane_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => seating_plane
connection_zone_interface_- plane_relationship to connection_zone (as mating_zone)	PATH			
connection_zone_interface_- plane_relationship to seating_plane (as interface_plane)	PATH			
GUIDED_WAVE_BARE_DIE_- TERMINAL	minimally_defined_bare_die_- terminal	210	333	minimally_defined_bare_die_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'guided wave bare die terminal'} minimally_defined_bare_die_terminal <= shape_aspect
MINIMALLY_DEFINED_- BARE_DIE_TERMINAL	minimally_defined_bare_die_- terminal	210	333	
minimally_defined_bare_die_- terminal to bare_die (as associated_definition)	PATH		333, 340, 338	minimally_defined_bare_die_terminal minimally_defined_bare_die_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => (physical_unit => bare_die) (physical_unit => externally_defined_physical_unit {externally_defined_physical_unit <= externally_defined_item} externally_defined_physical_unit => externally_defined_bare_die) (physical_unit => externally_defined_physical_unit {externally_defined_physical_unit <= externally_defined_item} externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die) </pre>
SEATING_PLANE	seating_plane	210		<pre> seating_plane <= shape_aspect {[shape_aspect shape_aspect.product_definitional = .FALSE.]} </pre>

Table 22 – Mapping table for bare_die UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
seating_plane to package (as scope)	PATH			<pre> shape_aspect shape_aspect ({shape_aspect (shape_aspect.name= 'package seating plane') (shape_aspect.name= 'interface plane')}) shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit => (externally_defined_physical_unit => (externally_defined_package) (library_defined_physical_unit => (library_defined_package))) (package)) ({shape_aspect (shape_aspect.name= 'die seating plane')}) shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => </pre>

Table 22 – Mapping table for bare_die UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				physical_unit => (externally_defined_physical_unit => (externally_defined_bare_die) (library_defined_physical_unit => (library_defined_bare_die))) (bare_die))

Table 23 – Mapping table for characteristic UoF

Application element	AIM element	Source	Rules	Reference path
ABSORBED_DOSE_DATA_ELEMENT	absorbed_dose_measure_with_unit	210		
ACTIVITY_DATA_ELEMENT	activity_measure_with_unit	210		
AMOUNT_OF_SUBSTANCE_DATA_ELEMENT	amount_of_substance_measure_with_unit	41		
ANGLE_DATA_ELEMENT	plane_angle_measure_with_unit	41		
AREA_DATA_ELEMENT	area_measure_with_unit	41		
CAPACITANCE_DATA_ELEMENT	capacitance_measure_with_unit	210		
CELSIUS_TEMPERATURE_DATA_ELEMENT	celsius_temperature_measure_with_unit	41		
CHARACTERISTIC	representation_item	43	31	<pre> representation_item { representation_item = representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} {group (group.name = 'numeric property')} </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(group.name = 'externally defined property') (group.name = 'compound characteristic') (group.name = 'boolean property') (group.name = 'logical property') (group.name = 'string property')}}
type_name	representation_item.name	43		
characteristic to ee_document (as reference_document)	PATH			representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'reference document'} document_reference document_reference.assigned_document -> (document) (document => ee_specification)
characteristic to boolean_property (as property_value)	PATH			representation_item {[document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11']}

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} group.name = 'boolean property'}] representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true')} (descriptive_representation_item.description = 'false')} </pre>
characteristic to data_element (as property_value)	PATH			<pre> representation_item => measure_representation_item {[measure_representation_item <= representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} group.name = 'numeric property'}]} </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[measure_representation_item <= measure_with_unit]]
characteristic to external_definition (as property_value)	PATH			<pre> representation_item => {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} group.name = 'externally defined property'} externally_defined_representation_item => external_definition representation_item {[representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11'} representation_item [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment.items[i] </pre>
characteristic to logical_property (as property_value)	PATH		109	<pre> {[representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11'} representation_item [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment.items[i] </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} group.name = 'logical property'}} representation_item => descriptive_representation_item.description {(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'unknown')}} representation_item => {[representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11'} [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] </pre>
characteristic to string_property (as property_value)	PATH			<pre> representation_item => {[representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11'} [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} group.name = 'string property'}} descriptive_representation_item descriptive_representation_item.description representation_item => {[representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= {document_reference document_reference.role -> object_role object_role.name = 'base type dictionary'} document_reference document_reference.assigned_document -> document document.id = 'ISO 10303-11'} [representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} </pre>
characteristic to compound_ characteristic_definition (as property_value)	PATH			

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> group.name = 'compound characteristic'} compound_representation_item ({compound_representation_item <= representation_item representation_item.name = 'list representation_item'}) compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item[i] -> representation_item) ({compound_representation_item <= representation_item representation_item.name = 'set representation item'}) compound_representation_item.item_element -> compound_item_definition = set_representation_item set_representation_item[i] -> representation_- item) representation_item </pre>
CHARACTERISTIC_DATA_SET	data_set_representation_item	210		<pre> data_set_representation_item <= compound_representation_item {compound_representation_item compound_representation_item.item_element -> compound_item_definition = set_- representation_item} compound_representation_item <= representation_item </pre>
CHARACTERISTIC_DATA_TA- BLE	table_representation_item	210		<pre> table_representation_item <= compound_representation_item {compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_- representation_item} </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				compound_representation_item <= representation_item
COMPOUND_CHARACTERIS- TIC	compound_representation_item	43		<pre> compound_representation_item {compound_representation_item <= representation_item} {(compound_representation_item => data_set_representation_item => (compound_representation_item => table_representation_item))} compound_representation_item <= applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group group.name = 'compound characteristic'} group => characteristic_type </pre>
compound_characteristic to characteristic (as property_value)	PATH			<pre> compound_representation_item ({compound_representation_item => table_representation_item} {compound_representation_item <= representation_item representation_item.name = 'list representation_item'}) compound_representation_item.item_element -> compound_item_definition = list_representation_item list_representation_item[i] -> representation_item) ({compound_representation_item => data_set_representation_item} {compound_representation_item <= representation_item </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item.name = 'set representation item' compound_representation_item.item_element -> compound_item_definition = set_representation_item set_representation_item[i] -> representation_- item) representation_item {group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type}} </pre>
CONDUCTANCE_DATA_ELEMENT	conductance_measure_with_unit	210		
COORDINATED_CHARACTERISTIC	coordinated_representation_item	210		<pre> coordinated_representation_item <= [representation] [representation_item] </pre>
dependent_environment	data_environment.name	45		<pre> coordinated_representation_item <= representation <- property_definition_representation.used_- representation property_definition_representation <- data_environment.elements[i] data_environment data_environment.name coordinated_representation_item <= representation representation.name </pre>
type_name	representation.name	43		<pre> coordinated_representation_item <= representation representation.name </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
coordinated_characteristic to characteristic (as component_characteristic)	PATH			coordinated_representation_item <= representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type}
coordinated_characteristic to ee_document (as reference_document)	PATH			coordinated_representation_item document_assigned_item = coordinated_ representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> (document) (document => ee_specification)
DATA_ELEMENT	measure_with_unit	41		
measure_value	PATH			measure_with_unit measure_with_unit.value_component
unit_of_measure	PATH			measure_with_unit measure_with_unit.unit_component -> unit = named_unit named_unit => (si_unit si_unit.name) (conversion_based_unit

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				conversion_based_unit.name) (context_dependent_unit context_dependent_unit.name)
unit_of_measure_prefix	si_unit.prefix	41		measure_with_unit measure_with_unit.unit.component -> unit = named_unit named_unit => si_unit si_unit.prefix {(si_unit.prefix = .exa.) (si_unit.prefix = .peta.) (si_unit.prefix = .tera.) (si_unit.prefix = .giga.) (si_unit.prefix = .mega.) (si_unit.prefix = .kilo.) (si_unit.prefix = .hecto.) (si_unit.prefix = .deca.) (si_unit.prefix = .deci.) (si_unit.prefix = .centi.) (si_unit.prefix = .milli.) (si_unit.prefix = .micro.) (si_unit.prefix = .nano.) (si_unit.prefix = .pico.) (si_unit.prefix = .femto.) (si_unit.prefix = .atto.)}}
DOSE_EQUIVALENT_DATA_- ELEMENT	dose_equivalent_measure_with_- unit	210		
EE_COLOUR	colour_rgb	46	148	
name	PATH	46		colour_rgb <= colour_specification colour_specification.name

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ELECTRIC_CHARGE_DATA_- ELEMENT	electric_charge_measure_with_- unit	210		
ELECTRIC_CURRENT_DATA_- ELEMENT	electric_current_measure_with_- unit	41		
ELECTROMOTIVE_FORCE_- DATA_ELEMENT	electromotive_force_measure_- with_unit	210		
ENERGY_DATA_ELEMENT	energy_measure_with_unit	210		
EVENT_DATA_ELEMENT	ratio_measure_with_unit	210		
EXPANDED_UNCERTAINTY	expanded_uncertainty	45		
coverage_factor	expanded_uncertainty.coverage_- factor	45		
FORCE_DATA_ELEMENT	force_measure_with_unit	210		
FREQUENCY_DATA_ELEMENT	frequency_measure_with_unit	210		
ILLUMINANCE_DATA_ELEMENT	illuminance_measure_with_unit	210		
INDUCTANCE_DATA_ELEMENT	inductance_measure_with_unit	210		
LENGTH_DATA_ELEMENT	length_measure_with_unit	41		
LUMINOUS_FLUX_DATA_ELEMENT	luminous_flux_measure_with_unit	210		
LUMINOUS_INTENSITY_- DATA_ELEMENT	luminous_intensity_measure_- with_unit	41		
MAGNETIC_FLUX_DATA_ELEMENT	magnetic_flux_measure_with_unit	210		
MAGNETIC_FLUX_DENSITY_- DATA_ELEMENT	magnetic_flux_density_measure_- with_unit	210		

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
MASS_DATA_ELEMENT	mass_measure_with_unit	41		
MAXIMUM_TOLERANCE_- CHARACTERISTIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item {representation_item representation_item.name = 'maximum tolerance'}]
maximum_tolerance_characteristic to_numeric_parameter (as component_characteristic)	PATH			coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
MEASURE_QUALIFICATION	measure_qualification	45		
name	measure_qualification.name	45		
description	measure_qualification.description	45		
measure_qualification to data_element (as qualified_measure)	PATH			measure_qualification measure_qualification.qualified_measure -> measure_with_unit
measure_qualification to value_qualifier (as qualifiers)	PATH			measure_qualification measure_qualification.qualifiers[i] -> (value_qualifier = precision_qualifier) (value_qualifier = type_qualifier) (value_qualifier = uncertainty_qualifier)
MINIMUM_TOLERANCE_- CHARACTERISTIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item {representation_item representation_item.name = 'minimum tolerance'}]

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
minimum_tolerance_characteristic to numeric_parameter (as component_characteristic)	PATH			coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
MODEL_PARAMETER	(model_parameter) ([model_parameter] [measure_representation_item]) ([model_parameter] [descriptive_representation_item])	210 210 41 210 210 43	109, 211	(model_parameter <= representation_item) ([model_parameter <= representation_item] [measure_representation_item <= representation_item]) ([model_parameter <= representation_item] [descriptive_representation_item <= representation_item] {representation_item group_assigned_item = representation_item group_assigned_item <= applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => parameter_type}
type_name	representation_item.name	43	109	model_parameter <= representation_item representation_item.name
unit_of_measure	measure_with_unit.unit_ component	41		model_parameter <= representation_item => measure_representation_item <= measure_with_unit measure_with_unit.unit_component
model_parameter to characteristic (as default_value)	PATH			model_parameter <= representation_item <= representation.items[i] representation <=

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_2 -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type} </pre>
model_parameter to coordinated_characteristic (as default_value)	PATH			<pre> model_parameter <= representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_1 -> representation => coordinated_representation_item </pre>
model_parameter to coordinated_characteristic (as valid_range)	PATH			<pre> model_parameter <= representation_item <- representation.items[i] representation <- </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'valid range'} representation_relationship.rep_1 -> representation representation.items[i] -> representation_item => coordinated_representation_item </pre>
model_parameter to ee_document (as reference_document)	PATH			<pre> model_parameter <= representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> (document) (document => ee_specification) </pre>
NUMERIC_PARAMETER	measure_representation_item	45		<pre> measure_representation_item <= representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group => characteristic_type} {group (group.name = 'numeric property')}} </pre>

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
numeric_parameter to data_element (as property_value)	PATH			measure_representation_item <= measure_with_unit
PARAMETER_ASSIGNMENT	parameter_assignment_-representation	210		parameter_assignment_representation <= representation
parameter_assignment to characteristic (as parameter_value)	PATH			parameter_assignment_representation <= representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type}
parameter_assignment to coordinated_characteristic (as parameter_value)	PATH			parameter_assignment_representation <= representation representation.items[i] -> representation_item => coordinated_representation_item
parameter_assignment to model_parameter (as assigned_parameter)	PATH			parameter_assignment_representation <= representation representation.items[i] -> representation_item => model_parameter
PITCH_CLASS	representation	43		{representation representation.name = 'pitch class'}
class_name	representation.description	43		representation.description
line_width_class to range_characteristic (as class_member)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'range characteristic'} representation_item => coordinated_representation_item

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PLUS_MINUS_TOLERANCE_- CHARACTERISTIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item {representation_item representation_item.name = 'plus minus tolerance'}]
plus_minus_tolerance_- characteristic to numeric_parameter (as component_characteristic)	PATH		109	coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
POWER_DATA_ELEMENT	power_measure_with_unit	210		
PRECISION_QUALIFIER	precision_qualifier	45		
precision_value	precision_qualifier.precision_value	45		
PRESSURE_DATA_ELEMENT	pressure_measure_with_unit	210		
QUALITATIVE_UNCERTAINTY	qualitative_uncertainty	45		
uncertainty_value	qualitative_- uncertainty.uncertainty_value	45		
RANGE_CHARACTERISTIC	coordinated_representation_item	210		coordinated_representation_item <= [representation] [representation_item {representation_item => descriptive_representation_item} {representation_item representation_item.name = 'range characteristic'}]
range_type	PATH			coordinated_representation_item <= representation_item => descriptive_representation_item descriptive_representation_item.description

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{(descriptive_representation_item.description = 'closed') (descriptive_representation_item.description = 'open') (descriptive_representation_item.description = 'lower_open') (descriptive_representation_item.description = 'upper_open')}
range_characteristic to numeric_parameter (as component_characteristic)	PATH			coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
RATIO_DATA_ELEMENT	ratio_measure_with_unit	41		
RESISTANCE_DATA_ELEMENT	resistance_measure_with_unit	210		
SCATTERING_PARAMETER_DATA_ELEMENT	scattering_parameter_measure_with_unit	210		
SOLID_ANGLE_DATA_ELEMENT	solid_angle_measure_with_unit	41		
STANDARD_UNCERTAINTY	standard_uncertainty	45		
uncertainty_value	standard_uncertainty.uncertainty_value	45		
STATISTICAL_TOLERANCE_CHARACTERISTIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item {representation_item representation_item.name = 'statistical tolerance'}]
SYMMETRICAL_TOLERANCE_CHARACTERISTIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item]

Table 23 – Mapping table for characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item representation_item.name = 'symmetrical tolerance'}}
symmetrical_tolerance_- characteristic to numeric_parameter (as component_characteristic)	PATH			coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
THERMODYNAMIC_TEM- PERATURE_DATA_ELEMENT	thermodynamic_temperature_- measure_with_unit	41		
TIME_DATA_ELEMENT	time_measure_with_unit	41		
TOLERANCE_CHARACTERIS- TIC	coordinated_representation_item	210	109	coordinated_representation_item <= [representation] [representation_item {representation_item (representation_item.name = 'tolerance') (representation_item.name = 'minimum tolerance') (representation_item.name = 'maximum tolerance') (representation_item.name = 'plus minus tolerance') (representation_item.name = 'symmetrical tolerance') (representation_item.name = 'statistical tolerance')}}] coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item
tolerance_characteristic to numeric_parameter (as component_characteristic)	PATH			

Table 23 – Mapping table for characteristic UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
TYPE_QUALIFIER	type_qualifier	45		
name	type_qualifier.name	45		
UNCERTAINTY_QUALIFIER	uncertainty_qualifier	45		
measure_name	uncertainty_qualifier.measure_name	45		
description	uncertainty_qualifier.description	45		
VOLUME_DATA_ELEMENT	volume_measure_with_unit	41		

Table 24 – Mapping table for connection_zone_to_analytical_model_port_mapping UoF

Application element	AIM element	Source	Rules	Reference path
CONNECTION_ZONE_- MODEL_PORT_ASSIGNMENT	property_definition_representation	41		property_definition_representation
connection_zone_model_port_- assignment to connection_zone (as source)	PATH			property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.description = 'connection zone'}
connection_zone_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port

Table 25 – Mapping table for constructive_solid_geometry UoF

Application element	AIM element	Source	Rules	Reference path
BLOCK	block	42		
BOOLEAN_CSG_RESULT	boolean_result	42		
csg_operator	boolean_operator	42		boolean_result boolean_result.operator -> boolean_operator {boolean_operator (boolean_operator = .UNION.) (boolean_operator = .INTERSECTION.) (boolean_operator = .DIFFERENCE.)}
boolean_csg_result to boolean_csg_result (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = boolean_result boolean_result
boolean_csg_result to boolean_csg_result (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = boolean_result boolean_result
boolean_csg_result to bound_volume_shape (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand (boolean_operand = half_space_solid) (boolean_operand = solid_model)
boolean_csg_result to bound_volume_shape (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand (boolean_operand = half_space_solid) (boolean_operand = solid_model)

Table 25 – Mapping table for constructive_solid_geometry UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
boolean_csg_result to csg_primitive (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = csg_primitive
boolean_csg_result to csg_primitive (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = csg_primitive
boolean_csg_result to primitive_2d (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = csg_primitive csg_primitive = primitive_2d
boolean_csg_result to primitive_2d (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = csg_primitive csg_primitive = primitive_2d
CSG_PRIMITIVE	geometric_representation_item	42		geometric_representation_item <= representation_item {representation_item.name = 'csg_primitive'}
CSG_SOLID	csg_shape_representation	515		
RIGHT_ANGULAR_WEDGE	right_angular_wedge	42		
RIGHT_CIRCULAR_CYLIN- DER	right_circular_cylinder	42		

Table 26 – Mapping table for datum_based_characteristic UoF

Application element	AIM element	Source	Rules	Reference path
DATUM_BASED_LENGTH_- MEASURE	representation	43	77	{representation (representation.name = 'maximum positive component height') (representation.name = 'maximum negative component height') (representation.name = 'maximum mounting clearance') (representation.name = 'minimum mounting clearance')}}
measure_orientation	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'descriptive orientation'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'normal') (descriptive_representation_item.description = 'reversed')}}
datum_based_length_measure to datum_plane (as of_datum)	PATH			representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition

Table 26 – Mapping table for datum_based_characteristic UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<code>property_definition.definition -></code> <code>characterized_definition</code> <code>characterized_definition = shape_definition</code> <code>shape_definition</code> <code>shape_definition = shape_aspect</code> <code>{shape_aspect</code> <code>shape_aspect.description = 'plane'}</code> <code>shape_aspect =></code> <code>physical_unit_datum</code>
datum_based_length_measure to length_data_element (as measure)	PATH		109	<code>representation</code> <code>representation.items[i] -></code> <code>{representation_item</code> <code>representation_item.name = 'measure'}</code> <code>representation_item =></code> <code>measure_representation_item <=</code> <code>measure_with_unit =></code> <code>length_measure_with_unit</code>
datum_based_length_measure to direction (as measure_orientation)	PATH			<code>representation</code> <code>representation.items[i] -></code> <code>{representation_item</code> <code>representation_item.name = 'geometric</code> <code>orientation'}</code> <code>representation_item =></code> <code>geometric_representation_item =></code> <code>direction</code>

Table 27 – Mapping table for datum_difference_based_characteristic UoF

Application element	AIM element	Source	Rules	Reference path
DATUM_DIFFERENCE	datum_difference	210		[datum_difference <= shape_aspect_relationship] [datum_difference <= shape_aspect] {shape_aspect_relationship [shape_aspect_relationship.related_shape_-aspect -> shape_aspect => physical_unit_datum] [shape_aspect_relationship.relatng_shape_-aspect -> shape_aspect => physical_unit_datum]}
name	descriptive_representation_- item.description	45		datum_difference <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape. element_relationship_characterization'} property_definition <- property_definition property_definition.representation.definition property_definition.representation property_definition.representation.used_- representation ->

Table 27 – Mapping table for datum_difference_based_characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.name = 'shape element relationship name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
datum_difference to datum (as precedent_datum)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_unit_datum </pre>
datum_difference to datum (as subsequent_datum)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum </pre>
DATUM_DIFFERENCE_- BASED_CHARACTERISTIC	datum_difference_based_- characteristic	210		<pre> datum_difference_based_characteristic <= representation_item </pre>
datum_difference_based_- characteristic to datum_difference (as of_datum_difference)	PATH			<pre> datum_difference_based_characteristic <= representation_item <- representation.items[i] representation <- { representation representation.name = 'of datum difference'} property_definition_representation.used_- representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 27 – Mapping table for datum_difference_based_characteristic UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => datum_difference

Table 28 – Mapping table for datum_difference_based_model UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_MODEL	analytical_model	210		analytical_model <= representation
name	PATH			analytical_model <= representation representation.name
analytical_model to ee_document (as reference_document)	PATH			analytical_model <= representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'reference document'}
analytical_model to analytical_model_definition (as associated_definition)	PATH			analytical_model <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition ->

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>property_definition</code> <code>property_definition.definition -></code> <code>characterized_definition</code> <code>characterized_definition = characterized_-</code> <code>product_definition</code> <code>characterized_product_definition</code> <code>characterized_product_definition = product_-</code> <code>definition</code> <code>product_definition =></code> <code>analytical_model_definition</code>
analytical_model to language_reference_manual (as representation_language)	PATH			<code>analytical_model <=</code> <code>representation</code> <code>document_assigned_item = representation</code> <code>document_assigned_item <-</code> <code>applied_document_reference.items[i]</code> <code>applied_document_reference <=</code> <code>document_reference</code> <code>document_reference.assigned_document -></code> <code>{document</code> <code>document.kind -></code> <code>document_type</code> <code>document_type.product_data_type = 'language</code> <code>reference_manual'}</code> <code>document =></code> <code>ee_specification</code>
analytical_model to ee_specification (as source_code)	PATH			<code>analytical_model <=</code> <code>representation</code> <code>document_assigned_item = representation</code> <code>document_assigned_item <-</code> <code>applied_document_reference.items[i]</code> <code>applied_document_reference <=</code> <code>document_reference</code> <code>document_reference.assigned_document -></code> <code>{document</code>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document.kind -> document_type document_type.product_data_type = 'source code'} document => ee_specification
analytical_model to model_parameter (as parameter)	PATH			analytical_model <= representation representation.items[i] -> representation_item => model_parameter
ANALYTICAL_MODEL_PORT	analytical_model_port	210	342, 5	analytical_model_port <= representation {representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition (characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect) (characterized_definition = characterized_ object characterized_object)}
distributed_properties	descriptive_representation_item	45	342	analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'distributed property'}

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = ,true') (descriptive_representation_item.description = ,false')} </pre>
nominal_signal_flow_direction	descriptive_representation_item	45		<pre> analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'nominal signal flow direction'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = ,input direction') (descriptive_representation_item.description = ,output direction') (descriptive_representation_item.description = ,bidirectional') (descriptive_representation_item.description = ,unknown direction') (descriptive_representation_item.description = ,not applicable')} </pre>
port_name	representation.name	43	342	<pre> analytical_model_port <= representation representation.name </pre>
port_type	descriptive_representation_item	45	342	<pre> analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'port type'} representation_item => </pre>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> descriptive_representation_item { descriptive_representation_item (descriptive_representation_item.description = 'string property type') (descriptive_representation_item.description = 'logical property type') (descriptive_representation_item.description = 'numeric property type') (descriptive_representation_item.description = 'physical property type') (descriptive_representation_item.description = 'boolean property type') }</pre>
unit_of_measure	(named_unit) (derived_unit)	41 41	342	<pre> analytical_model_port <= representation representation.context_of_items -> representation_context => global_unit_assigned_context global_unit_assigned_context.units[i] -> unit (unit = named_unit named_unit) (unit = derived_unit derived_unit)</pre>
analytical_model_port to analytical_model (as accessed_analytical_model)	PATH		342	<pre> analytical_model_port <= representation <- representation_relationship.rep_2 representation_relationship { representation_relationship (representation_relationship.name = 'access mechanism') representation_relationship.rep_1 -> representation => analytical_model</pre>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_MODEL_- SCALAR_PORT	analytical_model_port	210		<pre> analytical_model_port <= representation { group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group (group.name = 'scalar port') (group.name = 'digital scalar port')} </pre>
ANALYTICAL_MODEL_VEC- TOR_PORT	analytical_model_port	210	6, 342	<pre> representation { representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group (group.name = 'vector port') (group.name = 'digital vector port')} </pre>
size	measure_representation_item	45	342, 109, 105	<pre> analytical_model_port <= representation representation.items[i] -> representation_item representation_item.name = 'size' representation_item => measure_representation_item measure_representation_item <= measure_with_unit {measure_with_unit.unit_component -> unit </pre>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				unit = named_unit named_unit => context_dependent_unit} measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure
DATUM_DIFFERENCE	datum_difference	210		[datum_difference <= shape_aspect_relationship] [datum_difference <= shape_aspect] {shape_aspect_relationship [shape_aspect_relationship.related_shape_- aspect -> shape_aspect => physical_unit_datum] [shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_unit_datum]}
name	descriptive_representation_- item.description	45		datum_difference <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element relationship characterization'} property_definition <- property_definition_representation.definition property_definition_representation

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation -> representation {representation representation.name = 'shape element relationship name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => physical_unit_datum shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum </pre>
datum_difference to datum (as precedent_datum)	PATH			
datum_difference to datum (as subsequent_datum)	PATH			
DATUM_DIFFERENCE_ANALYTICAL_MODEL_PORT_ASSIGNMENT	datum_difference_analytical_ - model_port_assignment	210		<pre> datum_difference_analytical_model_port_ assignment <= property_definition_representation datum_difference_analytical_model_port_ assignment <= property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port </pre>
datum_difference_analytical_ - model_port_assignment to analytical_model_port (as assigned_analytical_model_port)	PATH			

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
datum_difference_analytical_model_port_assignment to datum_difference (as of_datum_difference)	PATH			datum_difference_analytical_model_port_assignment <= property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => datum_difference
DATUM_DIFFERENCE_BASED_MODEL_PARAMETER	datum_difference_based_model_parameter	210		datum_difference_based_model_parameter <= model_parameter <= representation_item
datum_difference_based_model_parameter to datum_difference (as of_datum_difference)	PATH			datum_difference_based_model_parameter <= model_parameter <= representation_item <= representation.items[i] representation <= {representation representation.name = 'of datum difference'} property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => datum_difference

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_DIFFERENCE_FUNCTIONAL_UNIT_USAGE_VIEW_TERMINAL_ASSIGNMENT	datum_difference_functional_unit_usage_view_terminal_assignment	210		datum_difference_functional_unit_usage_view_terminal_assignment <= shape_aspect_relationship
datum_difference_functional_unit_usage_view_terminal_assignment to datum_difference (as assigned_datum_difference)	PATH			datum_difference_functional_unit_usage_view_terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => datum_difference
datum_difference_functional_unit_usage_view_terminal_assignment to functional_unit_usage_view_terminal_definition (as assigned_functional_unit_definition)	PATH			datum_difference_functional_unit_usage_view_terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => functional_unit_terminal_definition
DIGITAL_ANALYTICAL_MODEL_PORT	analytical_model_port	210	342	analytical_model_port <= representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group {group.name = 'digital analytical model port'}
nominal_signal_flow_direction	descriptive_representation_item	45	342, 109	analytical_model_port <= representation representation.items[i] -> {representation_item representation_item.name = 'distributed property'}

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'input direction') (descriptive_representation_item.description = 'output direction') (descriptive_representation_item.description = 'bidirectional') (descriptive_representation_item.description = 'unknown direction')}} </pre>
LANGUAGE_REFERENCE_- MANUAL	ee_specification	210		<pre> ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'language reference manual'} </pre>
MODEL_PARAMETER	<pre> (model_parameter) ([model_parameter] [measure_representation_item]) ([model_parameter] [descriptive_representation_item]) </pre>	210 210 41 210 43	109, 211	<pre> (model_parameter <= representation_item) ([model_parameter <= representation_item] [measure_representation_item <= representation_item] ([model_parameter <= representation_item] [descriptive_representation_item <= representation_item]) {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment </pre>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				group_assignment.assigned_group -> group => parameter_type}
type_name	representation_item.name	43	109	model_parameter <= representation_item representation_item.name
unit_of_measure	measure_with_unit.unit_ component	41		model_parameter <= representation_item => measure_representation_item <= measure_with_unit measure_with_unit.unit_component
model_parameter to characteristic (as default_value)	PATH			model_parameter <= representation_item <- representation.items[i] representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_2 -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type}

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
model_parameter to coordinated_characteristic (as default_value)	PATH			<pre> model_parameter <= representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'default value'} representation_relationship.rep_1 -> representation => coordinated_representation_item </pre>
model_parameter to coordinated_characteristic (as valid_range)	PATH			<pre> model_parameter <= representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'valid range'} representation_relationship.rep_1 -> representation representation.items[i] -> representation_item => coordinated_representation_item </pre>
model_parameter to ee_document (as reference_document)	PATH			<pre> model_parameter <= representation_item document_assigned_item = representation_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> (document) </pre>

Table 28 – Mapping table for datum_difference_based_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(document => ee_specification)
TERMINAL_PAIR_PORT_- EQUIVALENCE_RELATION- SHIP	representation_relationship	43	359	{representation_relationship representation_relationship.name = 'terminal pair port equivalence relationship'}
terminal_pair_port_equivalence_- relationship to analytical_model_port (as equivalent_port)	PATH		342	representation_relationship representation_relationship.rep_2 -> {representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition (characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect) (characterized_definition = characterized_- object characterized_object))} representation => analytical_model_port
terminal_pair_port_equivalence_- relationship to analytical_model_scalar_port (as scalar_ports)	PATH		342	representation_relationship representation_relationship.rep_1 -> {representation group_assigned_item = representation group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <=

Table 28 – Mapping table for datum_difference_based_model UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>group_assignment group_assignment.assigned_group -> group (group.name = 'scalar port') (group.name = 'digital scalar port') [representation representation.name = 'analytical model scalar port'] [representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'scalar port list'} representation_relationship.rep_2 -> representation representation.name = 'analytical model scalar port'}} representation => analytical_model_port</pre>

Table 29 – Mapping table for design_activity_control UoF

Application element	AIM element	Source	Rules	Reference path
CHANGE_ORDER	directed_action	41	345, 332	{directed_action <= executed_action <= action <- action_assignment.assigned_action (change) (add_design_object_assignment) (delete_design_object_assignment) (change_to_design_object_assignment) (change_from_design_object_assignment) (rule_replacement_to_assignment) (rule_replacement_from_assignment)}
adopted_solution	action_method	41	345	directed_action <= executed_action <= action action.chosen_method -> action_method
change_order_to_design_object_- management_relationship (as adopted_solution)	PATH			directed_action <= executed_action <= action <- action_assignment.assigned_action (add_design_object_assignment) (delete_design_object_assignment) (change_to_design_object_assignment) (change_from_design_object_assignment) (rule_replacement_to_assignment) (rule_replacement_from_assignment)
CHANGE_REQUEST	versioned_action_request	41	334, 30	{versioned_action_request <- action_request_assignment.assigned_action_- request

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
consequence	action_method.consequence	41		action_request_assignment => change_request}
recommended_solution	action_method	41		versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method -> action_method action_method.consequence
version	versioned_action_request.version	41		versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method -> action_method
change_request to design_object_- management_relationship (as recommended_solution)	PATH			versioned_action_request <- action_request_assignment.assigned_action_- request action_request_assignment => (add_design_object_request_assignment) (delete_design_object_request_assignment) (change_to_design_object_request_assignment) (change_from_design_object_request_assignment)
START_ORDER	directed_action	41	345, 332	{directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment => start_work}
START_REQUEST	versioned_action_request	41	334	{versioned_action_request <- action_request_assignment.assigned_action_- request action_request_assignment => start_request}
WORK_ORDER	directed_action	41	97, 345, 332, 388, 389	{directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment =>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(change) (start_work)}
additional_data	action_directive.comment	41		directed_action directed_action.directive -> action_directive action_directive.comment
analysis_data	action_directive.analysis	41		directed_action directed_action.directive -> action_directive action_directive.analysis
work_order_id	action_directive.name	41		directed_action directed_action.directive -> action_directive action_directive.name
work_order to contract (as authorizing_agreement)	PATH			directed_action contract_assigned_item = directed_action contract_assigned_item <- applied_contract_assignment.items[i] applied_contract_assignment <= contract_assignment contract_assignment.assigned_contract -> contract
work_order to date (as release_date)	PATH		142, 102, 289, 101, 350	directed_action date_assigned_item = directed_action date_assigned_item <- applied_date_assignment.items[i] applied_date_assignment <= {date_assignment date_assignment.role -> date_role date_role.name = 'release date'} date_assignment date_assignment.assigned_date -> date => calendar_date

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
work_order to date (as start_date)	PATH			<pre> directed_action date_assigned_item = directed_action date_assigned_item <- applied_date_assignment.items[i] {date_assignment {date_assignment.role -> date_role date_role.name = 'start date'} date_assignment date_assignment.assigned_date -> date => calendar_date </pre>
work_order to date_and_time (as release_date)	PATH		142, 103	<pre> directed_action date_and_time_assigned_item = directed_action date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] {date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'release date'} date_and_time_assignment date_and_time_assignment.assigned_date_and_ time -> date_and_time </pre>
work_order to date_and_time (as start_date)	PATH			<pre> directed_action date_and_time_assigned_item = directed_action date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] {date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'start date'} </pre>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time
work_order to ee_approval (as work_order_approval)	PATH		141, 345, 13	directed_action <= executed_action <= action <= action_assignment.assigned_action action_assignment (action_assignment => start_work approval_assigned_item = start_work approval_assigned_item <= applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval) (action_assignment => change approval_assigned_item = change approval_assigned_item <= applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval)
work_order to ee_product_version (as resulting_product)	PATH			directed_action <= executed_action <= action <= action_assignment.assigned_action (action_assignment =>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> change.items[i] -> work_item work_item = product_definition_formation product_definition_formation (action_assignment => start_work start_work.items[i] -> work_item work_item = product_definition_formation product_definition_formation) </pre>
work_order to organization (as initiator)	PATH		106	<pre> directed_action organization_assigned_item = directed_action organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'initiator'} organization_assignment.assigned_organization -> organization </pre>
work_order to person_and_organization (as initiator)	PATH		108	<pre> directed_action person_and_organization_assigned_item = directed_action person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] </pre>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'initiator'} person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
work_order_to_work_request (as referenced_work_request)	PATH			<pre> directed action directed action.directive -> action_directive action_directive.requests[i] -> versioned_action_request </pre>
WORK_REQUEST	versioned_action_request	41	287, 334, 390, 391	<pre> {versioned_action_request <- action_request_assignment.assigned_action_- request action_request_assignment => (change_request) (start_request)} </pre>
description	versioned_action_- request.description	41		
reason	versioned_action_request.purpose	41		
status	action_request_status	41	383	<pre> versioned_action_request <- action_request_status.assigned_request action_request_status </pre>
work_request_id	versioned_action_request.id	41		
work_request_to evaluated_characteristic (as reason)	PATH			<pre> versioned_action_request <- action_request_assignment.assigned_action_- request action_request_assignment => </pre>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (change_request change_request.items[i] -> change_request_item change_request_item = representation_ relationship representation_relationship {representation_relationship.name = 'evaluated characteristic'}) (start_request start_request.items[i] -> start_request_item start_request_item = representation_ relationship representation_relationship {representation_relationship.name = 'evaluated characteristic'}) </pre>
work_request to date (as request_date)	PATH		381, 102, 289, 101, 350	<pre> versioned_action_request date_assigned_item = versioned_action_request date_assigned_item <- applied_date_assignment.items[i] applied_date_assignment <= {date_assignment date_assignment.role -> date_role date_role.name = 'request date'} date_assignment date_assignment.assigned_date -> date => calendar_date </pre>
work_request to date_and_time (as request_date)	PATH		103	<pre> versioned_action_request date_and_time_assigned_item = versioned_ action_request date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= </pre>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{ date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'request date' date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> time -> date_and_time }</pre>
work_request to ee_approval (as work_request_approval)	PATH		380, 334, 13	<pre>versioned_action_request <- action_request_assignment.assigned_action_request (action_request_assignment => start_request approval_assigned_item = start_request approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval) (action_request_assignment => change_request approval_assigned_item = change_request approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval)</pre>
work_request to ee_product_version (as referenced_product)	PATH			<pre>versioned_action_request <- action_request_assignment.assigned_action_request (action_request_assignment => change_request change_request.items[i] -> }</pre>

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				change_request_item change_request_item = product_definition_ - formation) (action_request_assignment => start_request start_request.items[i] -> start_request_item start_request_item = product_definition_ - formation)
work_request to organization (as initiator)	PATH		382, 106	versioned_action_request organization_assigned_item = versioned_ - action_request organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'initiator'} organization_assignment.assigned_organization -> organization
work_request to organization (as request_recipient)	PATH			versioned_action_request organization_assigned_item = versioned_ - action_request organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'request_recipient'}

Table 29 – Mapping table for design_activity_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				organization_assignment.assigned_organization -> organization
work_request to person_and_organization (as initiator)	PATH		382, 108	versioned_action_request person_and_organization_assigned_item = versioned_action_request person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'initiator'}
work_request to person_and_organization (as request_recipient)	PATH			person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization versioned_action_request person_and_organization_assigned_item = versioned_action_request person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'request recipient'}
				person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization

Table 29 – Mapping table for design_activity_control UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path

Table 30 – Mapping table for design_management UoF

Application element	AIM element	Source	Rules	Reference path
ADD_DESIGN_OBJECT_MANAGEMENT_RELATIONSHIP	(add_design_object_assignment) (add_design_object_request_assignment)	210 210	1	(add_design_object_assignment {add_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object addition'} <= action_assignment) (add_design_object_request_assignment {add_design_object_request_assignment.items[i] -> managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object addition'} <= action_request_assignment)
add_design_object_management_ relationship to managed_design_object (as current_design_object)	PATH			(add_design_object_assignment add_design_object_assignment.items[i] -> managed_design_object) (add_design_object_request_assignment add_design_object_request_assignment.items[i] -> managed_design_object)
CHANGE_DESIGN_OBJECT_ MANAGEMENT_RELATION- SHIP	(change_from_design_object_ assignment) (change_from_design_object_ request_assignment)	210 210	29	(change_from_design_object_assignment {change_from_design_object_assignment <= action_assignment} change_from_design_object_assignment.items[i] -> managed_design_object {managed_design_object

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object change' managed_design_object <- change_to_design_object_assignment.items[i] change_to_design_object_assignment {change_to_design_object_assignment <= action_assignment} (change_from_design_object_request_assignment {change_from_design_object_request_assignment <= action_request_assignment} change_from_design_object_request_ assignment.items[i] -> managed_design_object {managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object change' managed_design_object <- change_to_design_object_request_ assignment.items[i] change_to_design_object_request_assignment {change_to_design_object_request_assignment <= action_request_assignment}) (change_from_design_object_assignment.items[i] -> managed_design_object {managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object change' managed_design_object <- change_to_design_object_assignment.items[i]</pre>
change_design_object_ - management_relationship to managed_design_object (as current_design_object)	PATH			

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object (change_from_design_object_request_- assignment.items[i] -> managed_design_object {managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship.name = 'design object change'}) change_to_design_object_request_- assignment.items[i] change_to_design_object_request_assignment change_to_design_object_request_- assignment.items[i] -> managed_design_object </pre>
change_design_object_- management_relationship to managed_design_object (as previous_design_object)	PATH			<pre> (change_from_design_object_assignment change_from_design_object_assignment.items[i] -> managed_design_object) (change_from_design_object_request_assignment change_from_design_object_request_- assignment.items[i] -> managed_design_object) </pre>
DELETE DESIGN OBJECT_- MANAGEMENT_RELATION- SHIP	(delete_design_object_- assignment) (delete_design_object_request_- assignment)	210 210	96	<pre> (delete_design_object_assignment {delete_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship.name = 'design object deletion'}) <= </pre>

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> action_assignment (delete_design_object_request_assignment {delete_design_object_request_ assignment.items[i] -> managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship.name = 'design object deletion'} <= action_request_assignment) (delete_design_object_assignment delete_design_object_assignment.items[i] -> managed_design_object) (delete_design_object_request_assignment delete_design_object_request_ assignment.items[i] -> managed_design_object) </pre>
delete_design_object_ - management_relationship to managed_design_object (as previous_design_object)	PATH			
DESIGN_OBJECT_MANAGE- MENT_RELATIONSHIP	product_definition_relationship	41	127	<pre> {product_definition_relationship (product_definition_relationship.name = 'design_object_addition') (product_definition_relationship.name = 'design_object_change') (product_definition_relationship.name = 'design_object_deletion')} </pre>
design_object_management_ - relationship to ee_product_definition (as current_design)	PATH		338	<pre> product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => (product_definition_with_associated_documents) (physical_unit) (physical_unit => </pre>

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
design_object_management_- relationship to ee_product_definition (as previous_design)	PATH			product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition => (product_definition_with_associated_documents) (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
FROZEN DESIGN_OBJECT	frozen_assigned_item	210	353	frozen_assigned_item {(frozen_assigned_item = alternate_product_- relationship) (frozen_assigned_item = applied_date_and_- time_assignment)}

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(frozen_assigned_item = applied_document_ - reference) (frozen_assigned_item = applied_group_ - assignment) (frozen_assigned_item = characterized_object) (frozen_assigned_item = colour) (frozen_assigned_item = composite_curve_ - segment) (frozen_assigned_item = curve_style) (frozen_assigned_item = curve_style_font) (frozen_assigned_item = curve_style_font_and_ - scaling) (frozen_assigned_item = dimension_related_ - tolerance_zone_element) (frozen_assigned_item = dimensional_ - characteristic_representation) (frozen_assigned_item = dimensional_size) (frozen_assigned_item = document_usage_ - constraint) (frozen_assigned_item = externally_defined_ - item) (frozen_assigned_item = fill_area_style) (frozen_assigned_item = fill_area_style_ - colour) (frozen_assigned_item = geometric_tolerance) (frozen_assigned_item = geometric_tolerance_ - relationship) (frozen_assigned_item = group) (frozen_assigned_item = material_designation) (frozen_assigned_item = measure_with_unit) (frozen_assigned_item = plus_minus_tolerance) (frozen_assigned_item = pre_defined_item) (frozen_assigned_item = product_concept) (frozen_assigned_item = product_definition_ - relationship)

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(frozen_assigned_item = product_definition_formation_relationship) (frozen_assigned_item = property_definition) (frozen_assigned_item = property_definition_relationship) (frozen_assigned_item = property_definition_representation) (frozen_assigned_item = representation) (frozen_assigned_item = representation_relationship) (frozen_assigned_item = representation_context) (frozen_assigned_item = representation_item) (frozen_assigned_item = runout_zone_orientation) (frozen_assigned_item = shape_aspect) (frozen_assigned_item = shape_aspect_relationship) (frozen_assigned_item = tolerance_value) (frozen_assigned_item = tolerance_zone_form)}
MANAGED_DESIGN_OBJECT	managed_design_object	210	353	{managed_design_object (managed_design_object = product) (managed_design_object = alternate_product_relationship) (managed_design_object = product_concept) (managed_design_object = configuration_item) (managed_design_object = configuration_effectivity) (managed_design_object = product_definition) (managed_design_object = product_definition_relationship) (managed_design_object = product_definition_formation) (managed_design_object = product_definition_formation_relationship)}

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(managed_design_object = property_definition) (managed_design_object = property_definition_ - relationship) (managed_design_object = property_definition_ - representation) (managed_design_object = material_designation) (managed_design_object = characterized_object) (managed_design_object = shape_aspect) (managed_design_object = shape_aspect_ - relationship) (managed_design_object = representation) (managed_design_object = representation_ - relationship) (managed_design_object = representation_ - context) (managed_design_object = representation_item) (managed_design_object = composite_curve_ - segment) (managed_design_object = measure_with_unit) (managed_design_object = approval) (managed_design_object = approval_status) (managed_design_object = applied_approval_ - assignment) (managed_design_object = action) (managed_design_object = versioned_action_ - request) (managed_design_object = document) (managed_design_object = document_usage_ - constraint) (managed_design_object = applied_group_ - assignment) (managed_design_object = applied_document_ - reference) (managed_design_object = contract) (managed_design_object = date)

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(managed_design_object = applied_date_and_- time_assignment) (managed_design_object = certification) (managed_design_object = security_- classification) (managed_design_object = group) (managed_design_object = person_and_- organization) (managed_design_object = person) (managed_design_object = organization) (managed_design_object = dimensional_- characteristic_representation) (managed_design_object = dimensional_size) (managed_design_object = tolerance_zone_form) (managed_design_object = dimension_related_- tolerance_zone_element) (managed_design_object = geometric_tolerance) (managed_design_object = geometric_tolerance_- relationship) (managed_design_object = plus_minus_tolerance) (managed_design_object = tolerance_value) (managed_design_object = runout_zone_- orientation) (managed_design_object = externally_defined_- item) (managed_design_object = pre_defined_item) (managed_design_object = colour) (managed_design_object = curve_style) (managed_design_object = curve_style_font) (managed_design_object = curve_style_font_- and_scaling) (managed_design_object = fill_area_style) (managed_design_object = fill_area_style_- colour)}

Table 30 – Mapping table for design_management UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
OWNED_DESIGN_OBJECT	owner_assigned_item	210		{owner_assigned_item (owner_assigned_item = alternate_product_ - relationship) (owner_assigned_item = characterized_object) (owner_assigned_item = colour) (owner_assigned_item = component_definition) (owner_assigned_item = component_shape_aspect) (owner_assigned_item = composite_curve_ - segment) (owner_assigned_item = curve_style) (owner_assigned_item = curve_style_font) (owner_assigned_item = curve_style_font_and_ - scaling) (owner_assigned_item = dimension_related_ - tolerance_zone_element) (owner_assigned_item = dimensional_ - characteristic_representation) (owner_assigned_item = dimensional_size) (owner_assigned_item = document_usage_ - constraint) (owner_assigned_item = externally_defined_ - item) (owner_assigned_item = fill_area_style) (owner_assigned_item = fill_area_style_colour) (owner_assigned_item = geometric_tolerance) (owner_assigned_item = geometric_tolerance_ - relationship) (owner_assigned_item = group) (owner_assigned_item = inter_stratum_feature) (owner_assigned_item = material_designation) (owner_assigned_item = measure_with_unit) (owner_assigned_item = plus_minus_tolerance) (owner_assigned_item = pre_defined_item) (owner_assigned_item = product_concept) (owner_assigned_item = product_definition_ - formation_relationship)

Table 30 – Mapping table for design_management UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(owner_assigned_item = product_definition_- relationship) (owner_assigned_item = property_definition) (owner_assigned_item = property_definition_- relationship) (owner_assigned_item = property_definition_- representation) (owner_assigned_item = representation) (owner_assigned_item = representation_context) (owner_assigned_item = representation_item) (owner_assigned_item = representation_- relationship) (owner_assigned_item = runout_zone_- orientation) (owner_assigned_item = shape_aspect) (owner_assigned_item = shape_aspect_- relationship) (owner_assigned_item = stratum_feature) (owner_assigned_item = tolerance_value) (owner_assigned_item = tolerance_zone_form)}
OWNER_ASSIGNMENT	owner_assignment	210		owner_assignment
owned_item	PATH			owner_assignment owner_assignment.items
owner_assignment to person_and_organization (as assigned_owner)	PATH			owner_assignment <= person_and_organization_assignment

Table 31 – Mapping table for device_marking UoF

Application element	AIM element	Source	Rules	Reference path
CONFIGURATION_CONTROLLED_MESSAGE	descriptive_representation_item	45	60	
message_content	descriptive_representation_item.description	45		
message_type_specification	representation_item.name	43	109	descriptive_representation_item <= representation_item representation_item.name {(representation_item.name = 'approval date') (representation_item.name = 'manufacturer code') (representation_item.name = 'part number') (representation_item.name = 'revision identifier')}}
configuration_controlled_message to ee_approval (as message_source)	PATH		13	descriptive_representation_item <= representation_item <- representation.items[i] representation approval_assigned_item = representation approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
configuration_controlled_message to ee_product_version (as message_source)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_representation property_definition_representation

Table 31 – Mapping table for device_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.definition -> product_definition_formation </pre>
configuration_controlled_message to organization (as message_source)	PATH			<pre> descriptive_representation_item <= representation_item <- representation.items[i] representation organization_assigned_item = representation organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization </pre>
EE_SPECIFICATION	ee_specification	210		<pre> ee_specification <= document </pre>
ee_specification to organization (as source)	PATH			<pre> ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment </pre>

Table 31 – Mapping table for device_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> organization_assignment.role -> organization_role organization_role.name = 'document source' organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
MARKING	representation	43	206	<pre> {[representation representation.name = 'marking'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 31 – Mapping table for device_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect[]
marking to axis_placement (as placement)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'marking location'} representation_item => geometric_representation_item => placement
marking to ee_text (as marked_message)	PATH			representation representation.items[i] -> representation_item => descriptive_representation_item
marking to ee_material (as marking_material)	PATH			representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> {characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect} characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_ designation

Table 31 – Mapping table for device_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'marking material'} </pre>
marking to ee_specification (as marking_specification)	PATH			<pre> representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document <- document_usage_constraint.source document_usage_constraint document_usage_constraint.subject_element = 'marking specification'} document => ee_specification </pre>
marking to physical_unit_3d_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> representation {representation.name = '3d bound volume shape'} </pre>
marking to physical_unit_planar_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> </pre>

Table 31 – Mapping table for device_marking UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation {representation.name = 'planar projected shape'}
marking to feature_shape_definition (as message_representation)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'}
marking to styled_text (as message_representation_text)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'} representation_items[i] -> representation_item representation_item.name = 'marking text' representation_item => (geometric_representation_item => text_literal) (styled_item => annotation_occurrence => annotation_text_occurrence)

Table 32 – Mapping table for document UoF

Application element	AIM element	Source	Rules	Reference path
DESIGN_SPECIFICATION	ee_specification	210		ee_specification <= document {document.kind -> document_type document_type.product_data_type = 'design specification'}
EE_DOCUMENT	(document) (ee_specification)	41 210 150	149, 150	(document) (ee_specification <= document)
document_name	document.name	41		(ee_specification <= document) (document) document.name
revision	document.description	41		(ee_specification <= document) (document) document.description
ee_document to date_and_time (as change_date)	PATH			(ee_specification <= document) (document) document date_and_time_assigned_item = document date_and_time_assigned_item <= applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'change date'}

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assignment.assigned_date_and_time -> date_and_time
ee_document to date_and_time (as creation_date)	PATH			(ee_specification <= document) (document) document date_and_time_assigned_item = document date_and_time_assigned_item <= applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'creation date'} date_and_time_assignment.assigned_date_and_time -> time -> date_and_time
ee_document to ee_approval (as document_approval)	PATH		13	(ee_specification <= document) (document) document approval_assigned_item = document approval_assigned_item <= applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
ee_document to ee_document_identifier (as base_identifier)	PATH			(ee_specification <= document document_identifier_assigned_item = document) (document) document_identifier_assigned_item = document)

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document_identifier_assigned_item <- document_identifier_assignment.items[i] document_identifier_assignment <= group_assignment group_assignment.assigned_group -> group => document_identifier
ee_document to ee_text (as text_usage)	PATH			(ee_specification <= document <-) (document <-) document_reference.assigned_document document_reference => applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = descriptive_ representation_item descriptive_representation_item {descriptive_representation_item descriptive_representation_item.name = 'text usage'}
ee_document to structured_text (as key_phrase)	PATH			(ee_specification <= document <-) (document <-) document_reference.assigned_document document_reference => applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = representation {representation representation.name = 'key phrase'} representation => structured_text_representation_item

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_document to organization (as publisher)	PATH			<pre> (ee_specification <= document) (document) document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'publisher'} organization_assignment.assigned_organization -> organization </pre>
ee_document to person (as author)	PATH			<pre> (ee_specification <= document) (document) document person_assigned_item = document person_assigned_item <- applied_person_assignment.items[i] applied_person_assignment <= person_assignment {person_assignment person_assignment.role -> person_role person_role.name = 'author'} person_assignment.assigned_person -> person </pre>
ee_document to person_and_organization (as publisher)	PATH			<pre> (ee_specification <= document) (document) document </pre>

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'publisher'} person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
ee_document to security_classification (as security_code)	PATH			<pre> (ee_specification <= document) (document) document security_classification_assigned_item = document security_classification_assigned_item <- applied_security_classification_ assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
ee_document to structured_text (as text_usage)	PATH			<pre> (ee_specification <= document <-) (document <-) document.reference.assigned_document </pre>

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document_reference => applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = representation {representation representation.name = 'text usage'} representation => structured_text_representation_item
EE_DOCUMENT_IDENTIFIER	document_identifier	210		document_identifier <= group {group <- group_assignment.assigned_group group_assignment => document_identifier_assignment document_identifier_assignment.items[i] -> document_identifier_assigned_item = document}
assigned_value	group.description	41		document_identifier <= group group.description
identifier_name	group.name	41		document_identifier <= group group.name
EE_SPECIFICATION	ee_specification	210		ee_specification <= document
ee_specification to organization (as source)	PATH			ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role ->

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> organization_role organization_role.name = 'document source' organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization </pre>
EE_TEXT	descriptive_representation_item	45		
content	descriptive_representation_ - item.description	45		
FABRICATION_TECHNOL- OGY_SPECIFICATION	ee_specification	210		<pre> ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'fabrication technology specification'} </pre>

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERFACE_SPECIFICATION	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'interface specification'}}
LANGUAGE_REFERENCE_- MANUAL	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'language reference manual'}}
LEAD_FORM_SPECIFICATION	ee_specification	210		ee_specification <= document {[document document.kind -> document_type document_type.product_data_type = 'lead form specification'] [document < - document_relationship.related_document document_relationship document_relationship.relatng_document -> document document.kind -> document_type document_type.product_data_type = 'material specification']]}
MATERIAL_SPECIFICATION	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'material specification'}}

Table 32 – Mapping table for document UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PROCESS_SPECIFICATION	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'process specification'}
STRUCTURED_TEXT	structured_text_representation_- item	210		structured_text_representation_item <= [representation] [descriptive_representation_item <= representation_item] {representation representation.context_of_items -> representation.context => structured_text_representation.context}
structured_text to ee_text (as plain_text_content)	PATH			structured_text_representation_item <= representation representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
structured_text to external_definition (as content_markup)	PATH			structured_text_representation_item <= representation representation.items[i] -> representation_item => external_definition
structured_text to structured_text (as precedent_structured_text)	PATH			structured_text_representation_item <= representation <= representation.relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'precedent structured_text'} representation_relationship.rep_1 -> representation => structured_text_representation_item

Table 32 – Mapping table for document UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
structured_text to structured_text (as structured_text_content)	PATH			structured_text_representation_item <= representation representation.items[i] -> representation_item => descriptive_representation_item => structured_text_representation_item
SURFACE_FINISH_SPECIFICATION	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'surface finish specification'}
TEST_SPECIFICATION	ee_specification	210		ee_specification <= document {document document.kind -> document_type document_type.product_data_type = 'test specification'}

Table 33 – Mapping table for end_item_identification UoF

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT_CONFIGURATION	configuration_item	44	153, 154	
item_id	configuration_item.id	44		
phase_of_product	configuration_item.purpose	44		
ee_product_configuration to ee_approval (as product_configuration_approval)	PATH		13	configuration_item approval_assigned_item = configuration_item approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
ee_product_configuration to ee_product_version (as satisfying_product)	PATH			configuration_item <- configuration_design.configuration configuration_design configuration_design.design -> product_definition_formation
ee_product_configuration to ee_product_model (as model_of_product)	PATH		262	configuration_item configuration_item.item_concept -> product_concept
ee_product_configuration to organization (as configuration_manager)	PATH		106	configuration_item organization_assigned_item = configuration_ item organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <=

Table 33 – Mapping table for end_item_identification UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'configuration manager'} organization_assignment organization_assignment.assigned_organization -> organization </pre>
ee_product_configuration to person_and_organization (as configuration_manager)	PATH		61, 262, 108	<pre> configuration_item person_and_organization_assigned_item = configuration_item person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'configuration manager'} person_and_organization_assignment person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
EE_PRODUCT_MODEL	product_concept	44	158	
model_name	product_concept.id	44		

Table 34 – Mapping table for external_definition UoF

Application element	AIM element	Source	Rules	Reference path
EXTERNAL_DEFINITION	(document_usage_constraint) (external_definition) (bond_category)	41 210 210 210 210		(document_usage_constraint) (external_definition <= [descriptive_representation_item] [externally_defined_representation_item <= [externally_defined_item] [representation_item]] [characterized_object]) (bond_category <= [external_definition] [group])
code	(representation_item.name) (document_usage_- constraint.subject_element)	43 41		(external_definition <= descriptive_representation_item <= representation_item representation_item.name) (document_usage_constraint.subject_element)
code_value	(descriptive_representation_- item.description) (document_usage_- constraint.subject_element_value)	45 41		(external_definition <= descriptive_representation_item descriptive_representation_item.description) (document_usage_constraint.subject_element_- value)
external_definition to ee_document (as reference_document)	PATH			(external_definition <= characterized_object document_assigned_item = characterized_object) (bond_category document_assigned_item = bond_category) document_assigned_item <= applied_document_reference.items[i] applied_document_reference <= document_reference

Table 34 – Mapping table for external_definition UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				document_reference.assigned_document -> (document) (document => ee_specification)
external_definition to standard_ document_identification_string (as reference_document)	document.id			document_usage_constraint document_usage_constraint.source -> document document.id

Table 35 – Mapping table for fabrication_joint UoF

Application element	AIM element	Source	Rules	Reference path
CONNECTION_ZONE_- BASED_FABRICATION_JOINT	connection_zone_based_- fabrication_joint	210		connection_zone_based_fabrication_joint<= fabrication_joint <= shape_aspect
connection_zone_based_- fabrication_joint to connection_zone (as zone_1)	PATH			connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 1'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'}
connection_zone_based_- fabrication_joint to connection_zone (as zone_2)	PATH			connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 2'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'}
FABRICATION_JOINT	fabrication_joint	210		fabrication_joint <= [shape_aspect] [shape_aspect_relationship]
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal

Table 35 – Mapping table for fabrication_joint UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fabrication_joint to conductive_-interconnect_element_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect_element_terminal'} shape_aspect => component_terminal
fabrication_joint to conductive_-interconnect_element_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect_element_terminal'} shape_aspect => component_terminal
fabrication_joint to interconnect_module (as associated_interconnect_module)	PATH			fabrication_joint <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {product_definition.name = 'interconnect module'}
fabrication_joint to land_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'land join terminal'}

Table 35 – Mapping table for fabrication_join UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
fabrication_join to land_join_terminal (as fabrication_features_2)	PATH			fabrication_join <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
fabrication_join to non- functional_land_join_terminal (as fabrication_features_1)	PATH			fabrication_join <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal
fabrication_join to non- functional_land_join_terminal (as fabrication_features_2)	PATH			fabrication_join <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal

Table 35 – Mapping table for fabrication_joint UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fabrication_joint to printed_component_join_terminal (as fabrication_features_1)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to printed_component_join_terminal (as fabrication_features_2)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to stratum_- feature_template_component (as auxiliary_joint_material)	PATH		340, 338, 268, 333	<pre> fabrication_joint <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'auxiliary joint material'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {[shape_aspect.description = 'stratum feature template component']} [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 35 – Mapping table for fabrication_joint UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']}] shape_aspect => component_shape_aspect
fabrication_joint to via_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
fabrication_joint to via_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'}

Table 35 – Mapping table for fabrication_joint UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component_terminal'} shape_aspect => component_terminal
INTERCONNECT_MODULE_- PRODUCT_ASSEMBLY_- VIEW_FABRICATION_JOINT	sequential_laminate_passage_- based_fabrication_joint	210		sequential_laminate_passage_based_- fabrication_joint <= shape_aspect {shape_aspect.description = 'interconnect module product assembly view fabrication joint'}
interconnect_module_product_- assembly_view_fabrication_joint to interconnect_module_- component_stratum_based_- terminal (as joined_terminal)	PATH		333	sequential_laminate_passage_based_- fabrication_joint <= shape_aspect <- shape_aspect_relationship.relatiing_shape_- aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'interconnect module component stratum based terminal'} shape_aspect => component_terminal

Table 36 – Mapping table for form_features UoF

Application element	AIM element	Source	Rules	Reference path
BEVEL_EDGE_FEATURE_- SHAPE	shape_representation	41	111, 247	{shape_representation <= representation representation.name = 'bevel edge feature shape'}
bevel_edge_feature_shape to length_data_element (as maximum_bevel_width)	PATH		20, 111, 109	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'maximum bevel width'}
bevel_edge_feature_shape to length_data_element (as minimum_bevel_width)	PATH		20	representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
bevel_edge_feature_shape to angle_data_element (as maximum_bevel_angle)	PATH		20, 109	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'minimum bevel width'}
				representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
				shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'maximum bevel angle'}
				representation_item =>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit
bevel_edge_feature_shape to angle_data_element (as minimum_bevel_angle)	PATH		20	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'minimum bevel angle'} representation_item => measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit
BLIND_PASSAGE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'blind passage template'}
of_measure_orientation	descriptive_representation_item	45		part_template_definition <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'blind passage template measure orientation'} representation_item =>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> descriptive_representation_item { descriptive_representation_item (descriptive_representation_item.description = 'normal') (descriptive_representation_item.description = 'reversed') } </pre>
blind_passage_template to stratum_surface (as datum_reference)	PATH		333	<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect shape_aspect => stratum_surface </pre>
blind_passage_template to shape_dimension_representation (as bottom_distance_from_- datum_reference)	PATH			<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship => dimensional_location dimensional_characteristic = dimensional_- location dimensional_characteristic dimensional_characteristic <- dimensional_characteristic_- representation.dimension dimensional_characteristic_representation dimensional_characteristic_- representation.representation -> shape_dimension_representation </pre>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COUNTERBORE_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'counterbore passage template'}
counterbore_passage_template to unsupported_passage_template (as smaller_passage)	PATH		333, 340	part_template_definition <= shape_aspect <= shape_aspect <= shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect.relationship.description = 'smaller_passage'} shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition
counterbore_passage_template to blind_passage_template (as larger_passage)	PATH			part_template_definition <= shape_aspect shape_aspect <= shape_aspect <= shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect.relationship.description = 'larger_passage'} shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition
COUNTERSUNK_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'countersunk passage template'}

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
countersunk_passage_template to unsupported_passage_template (as constant_diameter_passage)	PATH		333, 340	<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'constant_diameter_passage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage_template'} shape_aspect => part_template_definition </pre>
countersunk_passage_template to blind_passage_template (as tapered_passage)	PATH			<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'larger_passage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage_template'} shape_aspect => part_template_definition </pre>
DATUM	physical_unit_datum	210	333	<pre> physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.} </pre>
EDGE_SEGMENT_CROSS_- SECTION	edge_segment_cross_section	210		<pre> {edge_segment_cross_section <= volume_shape_intersection <= [representation_relationship_with_- transformation representation_relationship_with_- transformation </pre>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship_with_- transformation.transformation_operator -> transformation [shape_representation = item_defined_transformation] [representation] [shape_representation_relationship <= representation_relationship representation_relationship.name = 'volume shape intersection']] {edge_segment_cross_section <= volume_shape_intersection <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition {property_definition property_definition.name = 'cross_section'}} property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition => inter_stratum_feature_edge_segment_template} </pre>
edge_segment_cross_section to edge_shape (as surface_intersection_result)	PATH			<pre> edge_segment_cross_section <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> </pre>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation representation.name = 'planar projected shape'} representation => shape_representation
edge_segment_cross_section to part_template_3d_shape (as projected_volume)	PATH			edge_segment_cross_section <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = '3d bound volume shape'} representation => shape_representation
EDGE_SHAPE	shape_representation	41		{shape_representation <= representation.context_of_items -> representation_context => geometric_representation_context.coordinate_- space_dimension = 2}
INTERCONNECT_MODULE_- EDGE_SEGMENT_SURFACE	interconnect_module_edge_- segment_surface	210		[interconnect_module_edge_segment_surface <= shape_aspect] [interconnect_module_edge_segment_surface <= shape_aspect_relationship] {shape_aspect.description = 'interconnect module edge segment surface'}
interconnect_module_edge_- segment_surface to interconnect_module_edge_- surface (as composed_surface)	PATH			interconnect_module_edge_segment_surface <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.related_shape_- aspect ->

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect shape_aspect.description = 'interconnect module edge surface'} shape_aspect</pre>
interconnect_module_edge_- segment_surface to edge_segment_vertex (as end_vertex)	PATH			<pre>interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex</pre>
interconnect_module_edge_- segment_surface to edge_segment_vertex (as start_vertex)	PATH			<pre>interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex</pre>
INTERCONNECT_MODULE_- PRIMARY_SURFACE	shape_aspect	41		<pre>shape_aspect {shape_aspect.description = 'interconnect module primary surface'}</pre>
INTERCONNECT_MODULE_- SECONDARY_SURFACE	shape_aspect	41		<pre>shape_aspect {shape_aspect.description = 'interconnect module secondary surface'}</pre>
INTERCONNECT_MODULE_- SURFACE_FEATURE	shape_aspect	41		<pre>shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface')}</pre>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}</pre>
interconnect_module_surface_- feature to interconnect_module_usage_view (as associated_definition)	PATH		194	<pre>shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']}] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)</pre>
RADIUS_EDGE_FEATURE_- SHAPE	shape_representation	41	111, 277, 247	<pre>{shape_representation <= representation representation.name = 'radius edge feature shape'}</pre>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
radius_edge_feature_shape to length_data_element (as maximum_radius_length)	PATH		111	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'maximum radius length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
radius_edge_feature_shape to length_data_element (as minimum_radius_length)	PATH			shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'minimum radius length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
STEP_EDGE_FEATURE_- SHAPE	shape_representation	41	111, 313, 247	{shape_representation <= representation representation.name = 'step edge feature shape'}
step_edge_feature_shape to length_data_element (as maximum_step_depth)	PATH		111, 109	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'maximum step depth'} representation_item =>

Table 36 – Mapping table for form_features UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <= measure_with_unit => length_measure_with_unit
step_edge_feature_shape to length_data_element (as maximum_step_width)	PATH			shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'maximum step width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
step_edge_feature_shape to length_data_element (as minimum_step_depth)	PATH		111, 109	shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'minimum step depth'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
step_edge_feature_shape to length_data_element (as minimum_step_width)	PATH			shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'minimum step width'} representation_item =>

Table 36 – Mapping table for form_features UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <= measure_with_unit => length_measure_with_unit

Table 37 – Mapping table for functional_decomposition_to_assembly_design_UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_SWAP_SPECIFICATION	[change_from_design_object_assignment] [change_to_design_object_assignment]	210 210		[change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment
component_swap_specification to design_functional_unit_- allocation_to_assembly_- component (as current_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'}
component_swap_specification to design_functional_unit_- allocation_to_assembly_- component (as previous_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_ relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'}

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_COMPOSITION_PATH	product_definition	41	114	product_definition {product_definition.id = 'design composition path'}
design_composition_path to functional_unit (as design_functional_unit)	PATH		338	product_definition product_definition <- product_definition_relationship.relationship-> product_definition product_definition_relationship {product_definition_relationship.name = 'design functional unit'}
				product_definition_relationship.relationship-> product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'}
				{product_definition <- product_definition_relationship.relationship-> product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit'] [product_definition_relationship.relationship-> product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')] [product_definition => (functional unit)]

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit[]}] product_definition => component_functional_unit product_definition product_definition <- product_definition_relationship.relatin_ - product_definition product_definition_relationship {product_definition_relationship.name = 'design path element'} product_definition_relationship.related_ - product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit </pre>
design_composition_path to functional_unit (as path_element)	PATH			
DESIGN_FUNCTIONAL_UNIT_ALLOCATION_TO_ASSEMBLY_COMPONENT	product_definition_relationship	41	115, 116	<pre> {product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to assembly component'} </pre>
global_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition </pre>

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} (descriptive_representation_item.description = 'true')} descriptive_representation_item </pre>
local_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_ - definition_relationship characterized_definition = characterized_ - product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item </pre>

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')} descriptive_representation_item
of_property	coordinated_representation_item	210		product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation => coordinated_representation_item
path_alias	product_definition_relationship.id			
swap_code	product_definition_- relationship.description			
design_functional_unit_- allocation_to_assembly_- component to assembly_component (as implementation)	PATH		338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition => component_definition)
design_functional_unit_- allocation_to_assembly_- component to design_composition_path (as functional_design_definition_- path)	PATH			product_definition_relationship {product_definition_relationship product_definition_relationship.description = 'design definition path'} product_definition_relationship.relation_- product_definition -> product_definition {product_definition.id = 'design composition path'}

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_FUNCTIONAL_- UNIT_ALLOCATION_TO_- ASSEMBLY_JOINT	product_definition_relationship	41	117, 118	{product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to assembly joint'}
design_functional_unit_- allocation_to_assembly_joint to assembly_joint (as implementation)	PATH		340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => assembly_joint
design_functional_unit_- allocation_to_assembly_joint to design_composition_path (as functional_design_definition_- path)	PATH			product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition.id = 'design composition path'}
GATE_PATH_SWAP_SPECIFI- CATION	{change_from_design_object_- assignment} [change_to_design_object_- assignment]	210 210		[change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment
gate_path_swap_specification to design_functional_unit_- allocation_to_reference_- functional_unit (as current_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to reference functional unit'}

Table 37 – Mapping table for functional_decomposition_to_assembly_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
gate_path_swap_specification to design_functional_unit_- allocation_to_reference_- functional_unit (as previous_design_object)	PATH			change_from_design_object_assignment change_from_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship.name = 'design functional unit allocation to reference functional unit'}
TERMINAL_SWAP_SPECIFICATION	[change_from_design_object_- assignment] [change_to_design_object_- assignment]	210 210		[change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment
terminal_swap_specification to functional_unit_terminal_node_- assignment (as current_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = property_definition_- relationship property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'functional unit terminal node assignment'}
terminal_swap_specification to functional_unit_terminal_node_- assignment (as previous_design_object)	PATH			change_from_design_object_assignment change_from_design_object_assignment.items[i] -> managed_design_object managed_design_object = property_definition_- relationship property_definition_relationship {property_definition_relationship.name = 'functional unit terminal node assignment'}

Table 38 – Mapping table for functional_decomposition_to_interconnect_design_UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_SWAP_SPECIFICATION	[change_from_design_object_- assignment] [change_to_design_object_- assignment]	210 210		[change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment
component_swap_specification to design_functional_unit_- allocation_to_assembly_- component (as current_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'}
component_swap_specification to design_functional_unit_- allocation_to_assembly_- component (as previous_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'}

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_COMPOSITION_PATH	product_definition	41	114	product_definition {product_definition.id = 'design composition path'}
design_composition_path to functional_unit (as design_functional_unit)	PATH		338	<pre> product_definition product_definition <- product_definition_relationship.relatiing_- product_definition product_definition_relationship {product_definition_relationship.name = 'design functional unit'} product_definition_relationship.related_- product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_- product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit']} [product_definition_relationship.relatiing_- product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_network_design') (application_context_element.name = 'functional design usage')] [product_definition => (functional_unit) </pre>

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit[]]) product_definition => component_functional_unit product_definition product_definition <- product_definition_relationship.relati- product_definition product_definition_relationship {product_definition_relationship.name = 'design path element'} product_definition_relationship.related_- product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit </pre>
design_composition_path to functional_unit (as path_element)	PATH			
DESIGN_FUNCTIONAL_-UNIT_ALLOCATION_TO_-ASSEMBLY_COMPONENT	product_definition_relationship	41	115, 116	<pre> {product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to assembly component'} </pre>
global_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition </pre>

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} (descriptive_representation_item.description = 'true')} descriptive_representation_item </pre>
local_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_ - definition_relationship characterized_definition = characterized_ - product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item </pre>

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')}
of_property	coordinated_representation_item	210		descriptive_representation_item product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation => coordinated_representation_item
path_alias	product_definition_relationship.id			
swap_code	product_definition_- relationship.description			
design_functional_unit_- allocation_to_assembly_- component to assembly_component (as implementation)	PATH		338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition => component_definition) ([product_definition => component_definition])

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect shape_aspect => component_shape_aspect]]</pre>
design_functional_unit_- allocation_to_assembly_- component to design_composition_path (as functional_design_definition_- path)	PATH			<pre>product_definition_relationship {product_definition_relationship product_definition_relationship.description = 'design definition path'} product_definition_relationship.relatin_ product_definition -> product_definition {product_definition.id = 'design composition path'}</pre>
DESIGN_FUNCTIONAL_- UNIT_ALLOCATION_TO_FAB- RICATION_JOINT	product_definition_relationship	41	119, 120	<pre>{product_definition_relationship product_definition_relationship.id = 'design functional unit allocation to fabrication joint'}</pre>
design_functional_unit_- allocation_to_fabrication_joint to fabrication_joint (as implementation)	PATH		340, 333	<pre>product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_- product_definition</pre>

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => fabrication_joint product_definition.relationship product_definition.relationship.relating_- product_definition -> product_definition {product_definition.id = 'design composition path'} </pre>
design_functional_unit - allocation_to_fabrication_joint to design_composition_path (as functional_design_definition_- path)	PATH			
GATE_PATH_SWAP_SPECIFI- CATION	[change_from_design_object_- assignment] [change_to_design_object_- assignment]	210 210		<pre> [change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment </pre>
gate_path_swap_specification to design_functional_unit_- allocation_to_reference_- functional_unit (as current_design_object)	PATH			<pre> change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to reference functional unit'} </pre>

Table 38 – Mapping table for functional_decomposition_to_interconnect_design_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
gate_path_swap_specification to design_functional_unit_- allocation_to_reference_- functional_unit (as previous_design_object)	PATH			change_from_design_object_assignment change_from_design_object_assignment.items[i] -> managed_design_object managed_design_object = product_definition_- relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to reference functional unit'} (component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
TERMINAL_SWAP_SPECIFI- CATION	[change_from_design_object_- assignment] [change_to_design_object_- assignment]	210 210		[change_from_design_object_assignment <=] [change_to_design_object_assignment <=] action_assignment
terminal_swap_specification to functional_unit_terminal_node_- assignment (as current_design_object)	PATH			change_to_design_object_assignment change_to_design_object_assignment.items[i] -> managed_design_object managed_design_object = property_definition_- relationship property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'functional unit terminal node assignment'}

Table 38 – Mapping table for functional_decomposition_to_interconnect_design_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
terminal_swap_specification to functional_unit_terminal_node_- assignment (as previous_design_object)	PATH			change_from_design_object_assignment change_from_design_object_assignment.items[i] -> managed_design_object managed_design_object = property_definition_- relationship property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'functional_unit_terminal_node_assignment'}
TRANSMISSION_LINE_FUNCTIONAL_UNIT_TERMINAL_- ALLOCATION	shape_aspect_relationship	41	375, 374	{shape_aspect_relationship shape_aspect_relationship.name = 'transmission line_functional_unit_terminal_allocation'}
transmission_line_functional_- unit_terminal_allocation to functional_unit_terminal (as transmission_line_functional_- unit_terminal)	PATH		333	shape_aspect_relationship.relation_shape_- aspect -> {shape_aspect shape_aspect.description = 'unit terminal'} shape_aspect => component_functional_terminal
transmission_line_functional_- unit_terminal_allocation to printed_component_join_terminal (as implementation)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal
transmission_line_functional_- unit_terminal_allocation to transmission_line_functional_- unit_to_routed_printed_- component_allocation (as functional_unit_design_path)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {[shape_aspect shape_aspect.description = 'printed component join terminal'] [shape_aspect => component_terminal]] shape_aspect <- shape_aspect_relationship.relation_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal in component'}

Table 38 – Mapping table for functional_decomposition_to_interconnect_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship {product_definition_relationship [product_definition_relationship.name = 'design_functional_unit_allocation']} [product_definition_relationship.description = 'transmission_line_to_routed_printed_component allocation'']] </pre>
TRANSMISSION_LINE_FUNCTIONAL_UNIT_TO_ROUTED_PRINTED_COMPONENT_ALLOCATION	product_definition_relationship	41	376	<pre> {product_definition_relationship [product_definition_relationship.name = 'design_functional_unit_allocation']} [product_definition_relationship.description = 'transmission_line_to_routed_printed_component allocation'']] </pre>
transmission_line_functional_unit_to_routed_printed_component_allocation_to_routed_printed_component (as implementation)	PATH		333	<pre> product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => routed_printed_component </pre>

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF

Application element	AIM element	Source	Rules	Reference path
BUS_ELEMENT_LINK	bus_element_link	210		{[bus_element_link <= product_definition_relationship] [bus_element_link <= product_definition]}
bus_element_link to bus_structural_definition (as precedent_element)	PATH			bus_element_link <= product_definition_relationship product_definition_relationship.related_ -> product_definition -> product_definition => bus_structural_definition
bus_element_link to bus_structural_definition (as subsequent_element)	PATH			bus_element_link <= product_definition_relationship product_definition_relationship.related_ -> product_definition -> product_definition => bus_structural_definition
bus_element_link to functional_ - unit_network_node_definition (as precedent_element)	PATH			bus_element_link <= product_definition_relationship product_definition_relationship.related_ -> product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}
bus_element_link to functional_ - unit_network_node_definition (as subsequent_element)	PATH			bus_element_link <= product_definition_relationship product_definition_relationship.related_ -> product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => network_node_definition
BUS_STRUCTURAL_DEFINITION	bus_structural_definition	210	23	bus_structural_definition <= product_definition
bus_name	product.name	41	268	bus_structural_definition <= product_definition product_definition.formation -> product_definition.formation product_definition.formation.of_product -> product product.name
bus_structural_definition to bus_element_link (as composition)	PATH			bus_structural_definition <= product_definition <- product_definition.relationship.relati- product_definition {product_definition.relationship product_definition.relationship.name = 'bus composition'} product_definition.relationship.related_ product_definition -> product_definition => bus_element_link
ELECTRICAL_NETWORK_DEFINITION	electrical_network	210	338, 268	electrical_network <= functional_unit <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}
FUNCTIONAL_UNIT	component_functional_unit	210	180	component_functional_unit <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <=

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element application_context_element.name = 'functional occurrence'
reference_designation	product_definition.id	41		component_functional_unit <= product_definition product_definition.id
functional_unit to parameter_assignment (as functional_property)	PATH			component_functional_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation <= representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'functional property'} representation_relationship.rep_2 -> representation => parameter_assignment_representation
functional_unit to functional_unit_definition (as definition)	PATH		338	component_functional_unit <= product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition_relationship product_definition_relationship.name = 'instantiated functional unit'} product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
functional_unit to functional_ unit_network_definition (as composed_network)	PATH			<pre> component_functional_unit <= product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'network composition'} product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element </pre>

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element.name = 'functional network design' product_definition => functional_unit </pre>
FUNCTIONAL_UNIT_DEFINITION	<pre> (functional_unit) (externally_defined_functional_ unit) (library_defined_functional_unit) </pre>	210 210 210	338	<pre> (functional_unit <= product_definition) (externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) (library_defined_functional_unit <= externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) {product_definition [product_definition.format -> product_definition.format product_definition.format.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'functionality'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')]} (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) </pre>
functional_unit_definition to analytical representation (as analytical_functional_unit - representation)	PATH			

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation => analytical_model} representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => analytical_representation
FUNCTIONAL_UNIT_NET- WORK_DEFINITION	functional_unit	210	338	functional_unit <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'} functional_unit <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'design usage'} product_definition_relationship
functional_unit_network_ - definition to functional_unit_usage_view (as usage_view)	PATH		338	

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional design usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
FUNCTIONAL_UNIT_NETWORK_NODE_DEFINITION	network_node_definition	210	338, 174	<pre> network_node_definition <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'} </pre>
functional_unit_network_node_ - name	product_definition.name	41	268, 338	<pre> network_node_definition <= product_definition product_definition.name </pre>
functional_unit_network_node_ - definition to functional_unit_network_ - definition (as associated_functional_unit_ - definition)	PATH		338	<pre> product_definition <= product_definition.relationship.related_ product_definition product_definition.relationship {product_definition.relationship product_definition.relationship.name = 'functional unit network node'} product_definition.relationship.related_ product_definition -> {product_definition </pre>

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network_design' product_definition => functional_unit
FUNCTIONAL_UNIT_NETWORK_TERMINAL_DEFINITION_BUS_ASSIGNMENT	property_definition_relationship	41	175	{property_definition_relationship.name = 'functional_unit_network_terminal_definition bus_assignment'}
functional_unit_network_- terminal_definition_bus_- assignment to bus_structural_definition (as connected_bus)	PATH			property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- characterized_definition = product_- product_definition characterized_product_definition = product_- definition product_definition => bus_structural_definition
functional_unit_network_- terminal_definition_bus_- assignment to functional_unit_usage_view_- terminal_definition (as connected_terminal)	PATH		333	property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => functional_unit_terminal_definition
FUNCTIONAL_UNIT_NETWORK_TERMINAL_DEFINITION_NODE_ASSIGNMENT	property_definition_relationship	41	176	{property_definition_relationship.name = 'functional unit network terminal definition node assignment'}
functional_unit_network_terminal_definition_node_assignment to functional_unit_network_node_definition (as composed_node)	PATH		338	property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition product_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}
functional_unit_network_terminal_definition_node_assignment to functional_unit_usage_view_terminal_definition (as connected_terminal)	PATH		333	property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition shape_definition = shape_definition shape_definition = shape_aspect shape_aspect

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => functional_unit_terminal_definition
FUNCTIONAL_UNIT_TERMINAL	component_functional_terminal	210	333	component_functional_terminal <= shape_aspect
functional_unit_terminal to functional_unit (as accessed_functional_unit)	PATH		333, 340, 338	component_functional_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit
functional_unit_terminal to functional_unit_usage_view_ terminal_definition (as definition)	PATH		333	component_functional_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated terminal'}

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => functional_unit_terminal_definition
FUNCTIONAL_UNIT_TERMINAL_BUS_ASSIGNMENT	property_definition_relationship	41	178	{property_definition_relationship.name = 'functional unit terminal bus assignment'}
functional_unit_terminal_bus_- assignment to bus_structural_definition (as connected_bus)	PATH			property_definition_relationship property_definition_relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => bus_structural_definition
functional_unit_terminal_bus_- assignment to functional_unit_terminal (as connected_terminal)	PATH		333	property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_functional_terminal
FUNCTIONAL_UNIT_TERMINAL_NODE_ASSIGNMENT	property_definition_relationship	41	179	{property_definition_relationship.name = 'functional unit terminal node assignment'}
functional_unit_terminal_node_- assignment to functional_unit_network_node_- definition (as composed_node)	PATH		338	property_definition_relationship property_definition_relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional network design'} product_definition => network_node_definition</pre>
functional_unit_terminal_node_- assignment to functional_unit_terminal (as connected_terminal)	PATH		333	<pre>property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_functional_terminal</pre>
FUNCTIONAL_UNIT_USAGE_- VIEW	<pre>(functional_unit) (externally_defined_functional_- unit) (library_defined_functional_unit)</pre>	210 210 210	338	<pre>(functional_unit <= product_definition) (externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) (library_defined_functional_unit <= externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) {product_definition product_definition.frame_of_reference -> product_definition_context <=</pre>

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element application_context_element.name = 'functional design usage'
FUNCTIONAL_UNIT_USAGE_- VIEW_TERMINAL_DEFINI- TION	functional_unit_terminal_- definition	210	181	functional_unit_terminal_definition <= shape_aspect
signal_name	shape_aspect.name	41	333	functional_unit_terminal_definition <= shape_aspect shape_aspect.name
functional_unit_usage_view_- terminal_definition to functional_unit_usage_view (as associated_functional_unit_- definition)	PATH		340, 338	shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional design usage'}
				product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
THERMAL_NETWORK_DEFI- NITION	thermal_network	210	338, 268	thermal_network <= functional_unit <= product_definition

Table 39 – Mapping table for functional_decomposition_with_nodal_representation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional network design' }

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF

Application element	AIM element	Source	Rules	Reference path
AGGREGATE_CONNECTIVITY_REQUIREMENT	product_definition_relationship	41	4	{product_definition_relationship product_definition_relationship.name = 'aggregate connectivity requirement'}
aggregate_connectivity_- requirement to functional_unit_network_node_- definition (as connectivity_requirement)	PATH		338	product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional network design'} product_definition => network_node_definition
aggregate_connectivity_- requirement to design_composition_path (as design_definition_path)	PATH		338	product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition product_definition.id = 'design composition path'}
aggregate_connectivity_- requirement to functional_unit_network_- definition (as design_definition_path)	PATH			product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {[product_definition => functional_unit] [product_definition

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}}
CONNECTIVITY_ALLOCATION	property_definition_relationship	41		{property_definition_relationship property_definition_relationship.name = 'connectivity allocation'}
connectivity_allocation to aggregate_connectivity_- requirement (as connectivity_requirement)	PATH			property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship {product_definition_relationship.name = 'aggregate connectivity requirement'}
connectivity_allocation to physical_connectivity_definition (as implementation)	PATH		340, 333	property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect => physical_connectivity_definition
DESIGN_COMPOSITION_PATH	product_definition	41	114	product_definition {product_definition.id = 'design composition path'}
design_composition_path to functional_unit (as design_functional_unit)	PATH		338	product_definition product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'design functional unit'}

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit']} [product_definition_relationship.related_ product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')] [product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit </pre>
design_composition_path to functional_unit (as path_element)	PATH			<pre> product_definition product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'design path element'} </pre>

Table 40 – Mapping table for functional_decomposition_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit </pre>
DESIGN_FUNCTIONAL_- UNIT_ALLOCATION_TO_- ASSEMBLY_COMPONENT	product_definition_relationship	41	115, 116	<pre> {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'} </pre>
global_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global_swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} </pre>

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'true')} descriptive_representation_item
local_swappable	descriptive_representation_item	43		product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')}} descriptive_representation_item
of_property	coordinated_representation_item	210		product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.used_ representation -> representation => coordinated_representation_item
path_alias	product_definition_relationship.id			
swap_code	product_definition_ relationship.description			
design_functional_unit_ allocation_to_assembly_ component to assembly_component (as implementation)	PATH		338, 340, 333	product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition => component_definition)
design_functional_unit_ allocation_to_assembly_ component to design_composition_path (as functional_design_definition_ path)	PATH			product_definition_relationship {product_definition_relationship product_definition_relationship.description = 'design_definition_path'} product_definition_relationship.relatin_ product_definition -> product_definition {product_definition.id = 'design composition path'}}
DESIGN_FUNCTIONAL_ UNIT_ALLOCATION_TO_REF_ ERENCE_FUNCTIONAL_UNIT	product_definition_relationship	41	121	{product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to reference functional unit'}
design_functional_unit_ allocation_to_reference_ functional_unit to design_composition_path (as design_definition_path)	PATH			product_definition_relationship product_definition_relationship.relatin_ product_definition -> product_definition {product_definition.id = 'design composition path'}}

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_functional_unit_- allocation_to_reference_- functional_unit to reference_composition_path (as reference_definition_path)	PATH			product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition.id = 'reference composition path'}
FUNCTIONAL_USAGE_- VIEW_TO_PART_TERMINAL_ ASSIGNMENT	shape_aspect_relationship	41	182, 183	{shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal allocation'}
functional_usage_view_to_part_- terminal_assignment to part_terminal (as physical_usage_view_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal => bare_die_terminal) (interconnect_module_terminal) (packaged_part_terminal)
functional_usage_view_to_part_- terminal_assignment to scalar_terminal_definition (as functional_usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relation_shape_- aspect -> {shape_aspect shape_aspect.description = 'scalar terminal'} shape_aspect => functional_unit_terminal_definition
MINIMALLY_DEFINED_COM- PONENT_TERMINAL	component_terminal	210		{component_terminal <= shape_aspect shape_aspect.description = 'minimally defined component terminal'}
minimally_defined_component_- terminal to part_terminal_external_reference (as definition)	PATH			component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'part terminal external reference'} representation </pre>
PACKAGE_EXTERNAL_REFERENCE	representation	43	220	<pre> {representation representation.name = 'package external reference'} </pre>
design_owner	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'design owner'} representation_item => descriptive_representation_item </pre>
part_number	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'part number'} representation_item => descriptive_representation_item </pre>
revision_code	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'revision code'} representation_item => descriptive_representation_item </pre>
product_definition_id	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'product definition id'} </pre>

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => descriptive_representation_item
PACKAGE_TERMINAL_EX- TERNAL_REFERENCE	descriptive_representation_item	45	221, 222	descriptive_representation_item <= {representation_item representation_item.name = 'package terminal external reference'} representation_item
name	descriptive_representation_- item.description	45		
package_terminal_external_- reference to package_external_reference (as associated_definition)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] {representation representation.name = 'package external reference'} representation
PART_TERMINAL_EXTER- NAL_REFERENCE	representation	45	236	{representation representation.name = 'part terminal external reference'} representation
feature_name	descriptive_representation_- item.description	41		representation.items[i] -> {representation_item representation_item.name = 'feature name'} representation_item => descriptive_representation_item descriptive_representation_item.description representation representation.items[i] ->
feature_type	descriptive_representation_- item.description	41		 {representation_item representation_item.name = 'feature type'} representation_item => descriptive_representation_item descriptive_representation_item.description

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_terminal_external_reference to physical_unit (as associated_definition)	PATH			<pre> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = characterized_ - product_definition characterized_product_definition = product_ - definition product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		<pre> (component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'assembly module component_terminal') (shape_aspect.description = 'bare die component_terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component_terminal') (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component_terminal') (shape_aspect.description = 'packaged component join terminal')}) </pre>

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(component_interface_terminal <= shape_aspect)
REFERENCE_COMPOSITION_- PATH	product_definition	41	278	product_definition {product_definition.id = 'reference composition path'}
reference_composition_path to functional_unit (as reference_functional_unit)	PATH		338	product_definition product_definition <- product_definition_relationship.relati- ng_ product_definition product_definition_relationship {product_definition_relationship.name = 'reference_functional_unit'} product_definition_relationship.related_ - product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_ - product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated_functional_unit'] [product_definition_relationship.relati- ng_ product_definition -> product_definition product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional_network_design') (application_context_element.name = 'functional_design_usage')]

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit</pre>
reference_composition_path to functional_unit (as path_element)	PATH			<pre>product_definition product_definition <- product_definition_relationship.relationship_ product_definition product_definition_relationship {product_definition_relationship.name = 'reference path element'} product_definition_relationship.related_ product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence'} product_definition => component_functional_unit</pre>
REFERENCE_FUNCTIONAL_ - UNIT_ASSIGNMENT_TO_PART	shape_aspect_relationship	41	279, 280	{shape_aspect_relationship shape_aspect_relationship.name = 'reference functional unit assignment to part'}
path_alias	shape_aspect_ - relationship.description	41		
swap_code	PATH			shape_aspect_relationship shape_definition = shape_aspect_relationship

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'swap code'} property_definition.description shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)
reference_functional_unit_ assignment_to_part to physical_unit (as implementation)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)
reference_functional_unit_ assignment_to_part to reference_composition_path (as functional_design_definition_ path)	PATH			shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.description = 'design_definition_path'} shape_aspect_relationship.related_shape_ aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 40 – Mapping table for functional_decomposition_with_nodal_representation_to_assembly_mapping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.id = 'reference composition path'}
reference_functional_unit_ assignment_to_part to coordinated_characteristic (as of_property)	PATH			shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => coordinated_representation_item

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF

Application element	AIM element	Source	Rules	Reference path
AGGREGATE_CONNECTIVITY_REQUIREMENT	product_definition_relationship	41	4	{product_definition_relationship product_definition_relationship.name = 'aggregate connectivity requirement'}
aggregate_connectivity_- requirement to functional_unit_network_node_- definition (as connectivity_requirement)	PATH		338	product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element network design'} product_definition => network_node_definition
aggregate_connectivity_- requirement to design_composition_path (as design_definition_path)	PATH		338	product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition product_definition.id = 'design composition path'}
aggregate_connectivity_- requirement to functional_unit_network_- definition (as design_definition_path)	PATH			product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {[product_definition => functional_unit] [product_definition

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}}
CONNECTIVITY_ALLOCATION	property_definition_relationship	41		{property_definition_relationship property_definition_relationship.name = 'connectivity allocation'}
connectivity_allocation to aggregate_connectivity_- requirement (as connectivity_requirement)	PATH			property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship {product_definition_relationship.name = 'aggregate connectivity requirement'}
connectivity_allocation to physical_connectivity_definition (as implementation)	PATH		340, 333	property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => physical_connectivity_definition
DESIGN_COMPOSITION_PATH	product_definition	41	114	product_definition {product_definition.id = 'design composition path'}
design_composition_path to functional_unit (as design_functional_unit)	PATH		338	product_definition product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'design functional unit'}

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit']} [product_definition_relationship.related_ product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')] [product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit </pre>
design_composition_path to functional_unit (as path_element)	PATH			<pre> product_definition product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'design path element'} </pre>

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit </pre>
DESIGN_FUNCTIONAL_- UNIT_ALLOCATION_TO_- ASSEMBLY_COMPONENT	product_definition_relationship	41	115, 116	<pre> {product_definition_relationship product_definition_relationship.name = 'design_functional_unit_allocation_to_assembly component'} </pre>
global_swappable	descriptive_representation_item	43		<pre> product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global_swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} </pre>

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'true')} descriptive_representation_item
local_swappable	descriptive_representation_item	43		product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')}} descriptive_representation_item
of_property	coordinated_representation_item	210		product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.used_ representation -> representation => coordinated_representation_item
path_alias	product_definition_relationship.id			
swap_code	product_definition_ relationship.description			
design_functional_unit_ allocation_to_assembly_ component to assembly_component (as implementation)	PATH		338, 340, 333	product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition => component_definition) ([product_definition => component_definition] [product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect shape_aspect => component_shape_aspect])
design_functional_unit_ allocation_to_assembly_ component to design_composition_path (as functional_design_definition_ path)	PATH			product_definition_relationship {product_definition_relationship product_definition_relationship.description = 'design_definition_path'} product_definition_relationship.relatin_ product_definition -> product_definition {product_definition.id = 'design composition path'}

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_FUNCTIONAL_UNIT_ALLOCATION_TO_REFERENCE_FUNCTIONAL_UNIT	product_definition_relationship	41	121	{product_definition_relationship product_definition_relationship.name = 'design functional unit allocation to reference functional unit'}
design_functional_unit_allocation_to_reference_functional_unit to design_composition_path (as design_definition_path)	PATH			product_definition_relationship product_definition_relationship.relationg_ - product_definition -> product_definition {product_definition.id = 'design composition path'}
design_functional_unit_allocation_to_reference_functional_unit to reference_composition_path (as reference_definition_path)	PATH			product_definition_relationship product_definition_relationship.related_ - product_definition -> product_definition {product_definition.id = 'reference composition path'}
FUNCTIONAL_USAGE_VIEW_TO_PART_TERMINAL_ASSIGNMENT	shape_aspect_relationship	41	182, 183	{shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal allocation'}
functional_usage_view_to_part_terminal_assignment to part_terminal (as physical_usage_view_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal => bare_die_terminal) (interconnect_module_terminal) (package_terminal) (packaged_part_terminal)

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
functional_usage_view_to_part_- terminal_assignment to scalar_terminal_definition (as functional_usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'scalar terminal'} shape_aspect => functional_unit_terminal_definition
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
REFERENCE_COMPOSITION_- PATH	product_definition	41	278	product_definition {product_definition.id = 'reference composition path'}
reference_composition_path to functional_unit (as reference_functional_unit)	PATH		338	product_definition product_definition <- product_definition_relationship.relatng_- product_definition product_definition_relationship {product_definition_relationship.name = 'reference functional unit'} product_definition_relationship.related_- product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'}

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated_functional_unit']} [product_definition_relationship.related_ product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional_network_design') (application_context_element.name = 'functional_design_usage')] [product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit </pre>
reference_composition_path to functional_unit (as path_element)	PATH			<pre> product_definition product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'reference_path_element'} </pre>

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit
REFERENCE_FUNCTIONAL_ UNIT_ASSIGNMENT_TO_PART	shape_aspect_relationship	41	279, 280	{shape_aspect_relationship shape_aspect_relationship.name = 'reference functional unit assignment to part'}
path_alias	shape_aspect_ relationship.description	41		
swap_code	PATH			shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'swap code'} property_definition.description
reference_functional_unit_ assignment_to_part to part_template (as implementation)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect => part_template_definition

Table 41 – Mapping table for functional_decomposition_with_nodal_representation_to_interconnect_mapping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
reference_functional_unit_- assignment_to_part to reference_composition_path (as functional_design_definition_- path)	PATH			<pre> shape_aspect_relationship { shape_aspect_relationship { shape_aspect_relationship.description = 'design definition path' } shape_aspect_relationship.relation_shape_- aspect -> shape_aspect { shape_aspect.of_shape -> product_definition_shape <= property_definition { property_definition.definition -> characterized_definition { characterized_definition = characterized_- product_definition { characterized_product_definition = product_- definition { product_definition { product_definition.id = 'reference composition path' } } } } } } } </pre>
reference_functional_unit_- assignment_to_part to coordinated_characteristic (as of_property)	PATH			<pre> shape_aspect_relationship { shape_definition = shape_aspect_relationship { characterized_definition = shape_definition { characterized_definition <- property_definition.definition { property_definition <- { property_definition.representation.definition { property_definition.representation { property_definition.representation.used_- representation -> representation { representation.items[i] -> representation_item => coordinated_representation_item } } } } } } } } </pre>

Table 42 – Mapping table for functional_unit_usage_definition to physical_unit_usage_view_mapping UoF

Application element	AIM element	Source	Rules	Reference path
FUNCTIONAL_USAGE_- VIEW_TO_PART_TERMINAL_ ASSIGNMENT	shape_aspect_relationship	41	182, 183	{shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal_allocation'}
functional_usage_view_to_part_- terminal_assignment to part_terminal (as physical_usage_view_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal => bare_die_terminal) (interconnect_module_terminal) (package_terminal) (packaged_part_terminal) shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'scalar terminal'} shape_aspect => functional_unit_terminal_definition
functional_usage_view_to_part_- terminal_assignment to scalar_terminal_definition (as functional_usage_view_terminal)	PATH			product_definition {product_definition.id = 'reference composition path'} product_definition <- product_definition relationship.relatng_- product_definition product_definition relationship {product_definition_relationship.name = 'reference functional unit'}
REFERENCE_COMPOSITION_- PATH	product_definition	41	278	
reference_composition_path to functional_unit (as reference_functional_unit)	PATH		338	

Table 42 – Mapping table for functional_definition_to_physical_unit_usage_view_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit']} [product_definition_relationship.related_ product_definition -> product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')] [product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit </pre>
reference_composition_path to functional_unit (as path_element)	PATH			<pre> product_definition product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'reference path element'} </pre>

Table 42 – Mapping table for functional_definition_to_physical_unit_usage_view_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit
REFERENCE_FUNCTIONAL_ UNIT_ASSIGNMENT_TO_PART	shape_aspect_relationship	41	279, 280	{shape_aspect_relationship shape_aspect_relationship.name = 'reference functional unit assignment to part'}
path_alias	shape_aspect_ relationship.description	41		
swap_code	PATH			shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'swap code'}
reference_functional_unit_ assignment_to_part to physical_unit (as implementation)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <=

Table 42 – Mapping table for functional_definition_to_physical_unit_usage_view_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit) </pre>
reference_functional_unit_ - assignment_to_part to reference_composition_path (as functional_design_definition_ - path)	PATH			<pre> shape_aspect_relationship {shape_aspect_relationship.description = shape_aspect_relationship.path'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition </pre>

Table 42 – Mapping table for functional_definition_to_physical_unit_usage_view_mapping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.id = 'reference composition path'}
reference_functional_unit_ assignment_to_part to coordinated_characteristic (as of_property)	PATH			shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => coordinated_representation_item

Table 43 – Mapping table for functional_definition_to_printed_part_mapping UoF

Application element	AIM element	Source	Rules	Reference path
FUNCTIONAL_USAGE_- VIEW_TO_PRINTED_PART_- TEMPLATE_TERMINAL_AS- SIGNMENT	shape_aspect_relationship	41	184, 185	{shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal allocation'}
functional_usage_view_to_- printed_part_template_terminal_- assignment to printed_part_template_terminal (as physical_usage_view_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')} shape_aspect => printed_part_template_terminal
functional_usage_view_to_- printed_part_template_terminal_- assignment to scalar_terminal_definition (as functional_usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'scalar terminal'} shape_aspect => functional_unit_terminal_definition
REFERENCE_COMPOSITION_- PATH	product_definition	41	278	product_definition {product_definition.id = 'reference composition path'}
reference_composition_path to functional_unit (as reference_functional_unit)	PATH		338	product_definition product_definition <- product_definition_relationship.relatng_- product_definition product_definition_relationship {product_definition_relationship.name = 'reference functional unit'}

Table 43 – Mapping table for functional_definition_to_printed_part_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional occurrence'} {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship [product_definition_relationship.name = 'instantiated functional unit']} [product_definition_relationship.related_ product_definition -> product_definition product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')] [product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit]]} product_definition => component_functional_unit </pre>
reference_composition_path to functional_unit (as path_element)	PATH			<pre> product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'reference path element'} </pre>

Table 43 – Mapping table for functional_definition_to_printed_part_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit </pre>
REFERENCE_FUNCTIONAL_ - UNIT_ASSIGNMENT_TO_PART	shape_aspect_relationship	41	279, 280	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'reference functional unit assignment to part'} </pre>
path_alias	shape_aspect_ - relationship.description	41		
swap_code	PATH			<pre> shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'swap code'} property_definition.description </pre>
reference_functional_unit_ - assignment_to_part to part_template (as implementation)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect => part_template_definition </pre>

Table 43 – Mapping table for functional_definition_to_printed_part_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
reference_functional_unit_- assignment_to_part to physical_unit (as implementation)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)
reference_functional_unit_- assignment_to_part to reference_composition_path (as functional_design_definition_- path)	PATH			shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.description = 'design_definition_path'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 43 – Mapping table for functional_definition_to_printed_part_mapping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.id = 'reference composition path'} product_definition </pre>
reference_functional_unit_- assignment_to_part to coordinated_characteristic (as of_property)	PATH			<pre> shape_aspect_relationship shape_definition = shape_aspect_relationship characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => coordinated_representation_item </pre>

Table 44 – Mapping table for functional_design_elaboration UoF

Application element	AIM element	Source	Rules	Reference path
MINIMALLY_DEFINED_COMPONENT_TERMINAL	component_terminal	210		{component_terminal <= shape_aspect shape_aspect.description = 'minimally defined component terminal'}
minimally_defined_component_terminal to part_terminal_external_reference (as definition)	PATH			component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition property_definition <- property_definition.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'part terminal external reference'}
PART_TERMINAL_EXTERNAL_REFERENCE	representation	45	236	{representation representation.name = 'part terminal external reference'}
feature_name	descriptive_representation_item.description	41		representation.items[i] -> {representation_item representation_item.name = 'feature name' representation_item => descriptive_representation_item.description

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
feature_type	descriptive_representation_- item.description	41		representation representation.items[i] -> {representation_item representation_item.name = 'feature type'} representation_item => descriptive_representation_item descriptive_representation_item.description
part_terminal_external_reference to part_template (as associated_definition)	PATH			representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'assembly module component_terminal') (shape_aspect.description = 'bare die component_terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal')

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'packaged component join terminal')}) (component_interface_terminal <= shape_aspect)
PHYSICAL_CONNECTIVITY_- DEFINITION	physical_connectivity_definition	210	333	physical_connectivity_definition <= shape_aspect
name	shape_aspect.name	41	333	physical_connectivity_definition <= shape_aspect shape_aspect.name
physical_connectivity_definition to physical_unit_network_definition (as associated_definition)	PATH			physical_connectivity_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => physical_unit_network_definition physical_connectivity_definition <= shape_aspect <- shape_aspect.relationship.relatng_shape_ aspect shape_aspect.relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated terminals'}
physical_connectivity_definition to physical_component_terminal (as associated_terminals)	PATH			

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal') } shape_aspect => component_terminal </pre>
PHYSICAL_CONNECTIVITY_- DEFINITION_RELATIONSHIP	shape_aspect_relationship	41	240	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'physical connectivity definition change'} </pre>
physical_connectivity_definition_- relationship to ee_requirement_occurrence (as requirement)	PATH			<pre> shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'requirement'} property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'requirement'} property_definition_relationship.related_- property_definition -> </pre>

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition => requirements_property
physical_connectivity_definition_- relationship to physical_connectivity_definition (as current_definition)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition
physical_connectivity_definition_- relationship to physical_connectivity_definition (as previous_definition)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition
PHYSICAL_CONNECTIVITY_- ELEMENT	physical_connectivity_element	210	241	physical_connectivity_element <= [shape_aspect] [shape_aspect_relationship]
name	shape_aspect.name	41	333	physical_connectivity_element <= shape_aspect shape_aspect.name
physical_connectivity_element to component_terminal (as end_terminus)	PATH		344, 333	physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal')}

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> } shape_aspect => component_terminal physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal') (shape_aspect.description = 'minimally defined component terminal') } shape_aspect => component_terminal </pre>
physical_connectivity_element to component_terminal (as start_terminus)	PATH		344, 333	
physical_connectivity_element to physical_connectivity_structure_- definition (as connectivity_context)	PATH			<pre> physical_connectivity_element <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'structure element'} shape_aspect_relationship </pre>

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => physical_connectivity_definition </pre>
physical_connectivity_element to topological_junction (as end_terminus)	PATH		344	<pre> physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'topological junction'} </pre>
physical_connectivity_element to topological_junction (as start_terminus)	PATH			<pre> physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {shape_aspect shape_aspect.description = 'topological junction'} </pre>
PHYSICAL_CONNECTIVITY_- MAKE_FROM_RELATIONSHIP	make_from_connectivity_- relationship	210		<pre> make_from_connectivity_relationship <= shape_aspect_relationship </pre>
physical_connectivity_make_- from_relationship to physical_connectivity_definition (as resultant_definition)	PATH		344, 333	<pre> make_from_connectivity_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition </pre>

Table 44 – Mapping table for functional_design_elaboration UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_connectivity_make_- from_relationship to physical_connectivity_definition (as reusable_definition)	PATH			make_from_connectivity_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition
PHYSICAL_CONNECTIVITY_- STRUCTURE_DEFINITION	physical_connectivity_definition	210	333	physical_connectivity_definition <= shape_aspect
PHYSICAL_UNIT_NET- WORK_DEFINITION	physical_unit_network_definition	210		physical_unit_network_definition <= product_definition {product_definition product_definition.formatation -> product_definition.formatation product_definition.formatation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'physical network design'}
TOPOLOGICAL_JUNCTION	shape_aspect	41	372, 373	{shape_aspect [shape_aspect.description = 'topological junction'] [shape_aspect.product_definitional = .TRUE.]}
name	shape_aspect.name	41		
topological_junction to physical_- connectivity_structure_definition (as scope)	PATH		333	shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'topological junction scope'} shape_aspect_relationship.relatng_shape_- aspect ->

Table 44 – Mapping table for functional_design_elaboration UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => physical_connectivity_definition

Table 45 – Mapping table for functional_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
FUNCTIONAL_UNIT_REQUIREMENT_ALLOCATION	product_definition_relationship	41	177	{product_definition_relationship product_definition_relationship.name = 'functional unit requirement allocation'}
functional_unit_requirement_- allocation to ee_requirement_occurrence (as functional_requirement)	PATH			product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement'} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'functional requirement'} property_definition.relationship.relatiing_- property_definition -> property_definition => requirements_property

Table 45 – Mapping table for functional_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
functional_unit_requirement_- allocation to functional_unit (as functional_unit_assignment)	PATH			product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit

Table 46 – Mapping table for functional_specification UoF

Application element	AIM element	Source	Rules	Reference path
EQUIVALENT_FUNCTIONAL_UNIT_DEFINITION_ASSIGNMENT	product_definition	41	165	{product_definition product_definition.description = 'equivalent functional unit definition'}
name	product_definition.id	41		
equivalent_functional_unit_definition_assignment_to_functional_unit_definition (as equivalent_functional_unit_definitions)	PATH		338	product_definition <- product_definition.relationship.relati- product_definition product_definition.relationship {product_definition_relationship product_definition_relationship.name = 'equivalent functional unit definition assignment'} product_definition.relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'functional network design') (application_context_element.name = 'functional design usage')} product_definition => functional_unit
FUNCTIONAL_SPECIFICATION	functional_specification	210		functional_specification <= representation
functional_specification_to_external_definition (as characteristic_category)	PATH			functional_specification <= representation <- representation_relationship.rep_2 {representation_relationship

Table 46 – Mapping table for functional_specification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.name = 'functional characteristic category' representation_relationship.rep_1 -> representation representation.items[i] -> representation_item => descriptive_representation_item => external_definition </pre>
functional_specification to functional_specification_definition (as associated_definition)	PATH			<pre> functional_specification <= representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition => functional_unit => functional_specification_definition </pre>
functional_specification to functional_unit (as characterizing_functional_unit)	PATH			<pre> functional_specification <= representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 46 – Mapping table for functional_specification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'} product_definition => component_functional_unit
functional_specification to signal_specification (as characterizing_signal)	PATH			functional_specification <= representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'characterizing_signal for functional specification'} representation_relationship.rep_2 -> representation {representation representation.name = 'signal_specification'} functional_specification <= representation <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'reference signal for functional_specification'} representation_relationship.rep_2 -> representation {representation representation.name = 'signal_specification'}
functional_specification to signal_specification (as reference_signal)	PATH			

Table 46 – Mapping table for functional_specification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
FUNCTIONAL_SPECIFICATION_DEFINITION	functional_specification_definition	210		functional_specification_definition <= functional_unit <= product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional network design'}
SIGNAL_SPECIFICATION	representation	43	340, 311	{representation representation.name = 'signal specification'}
signal_specification to functional_unit (as characterizing_functional_unit)	PATH		340, 338	representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = characterized_definition product_definition characterized_product_definition = product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional occurrence'}
signal_specification to functional_unit_network_definition (as characterization_network)	PATH			product_definition => component_functional_unit representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition

Table 46 – Mapping table for functional_specification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element network_design'} product_definition => functional_unit </pre>
signal_specification to functional_unit_terminal (as measurement_access_mechanism)	PATH			<pre> {representation representation.items[i] -> representation_item representation_item.name = 'measurement access mechanism'} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_functional_terminal </pre>

Table 46 – Mapping table for functional_specification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
signal_specification to functional_unit_terminal (as reference_access_mechanism)	PATH			<pre> {representation representation.items[i] -> representation_item representation_item.name = 'reference access mechanism'} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_functional_terminal }</pre>
signal_specification to signal_property (as maximum_characteristic)	PATH		109	<pre> {representation representation.items[i] -> representation_item representation_item.name = 'maximum characteristic'} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition {property_definition property_definition.name = 'signal property'} }</pre>
signal_specification to signal_property (as minimum_characteristic)	PATH			<pre> {representation representation.items[i] -> representation_item representation_item.name = 'minimum characteristic'} }</pre>

Table 46 – Mapping table for functional_specification UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition {property_definition property_definition.name = 'signal property'}
signal_specification to test_specification (as measurement_condition)	PATH		104	representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'test specification'} document => ee_specification

Table 47 – Mapping table for functional_unit_analytical_representation UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_MODEL_- PORT_ASSIGNMENT	property_definition_representation	41		
analytical_model_port_assignment to analytical_model_port (as assigned_analytical_model_port)	PATH		342	property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port
analytical_model_port_assignment to functional_unit_usage_view_- terminal_definition (as assigned_functional_unit_- terminal)	PATH			property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => functional_unit_terminal_definition
ANALYTICAL_REPRESENTA- TION	analytical_representation	210		analytical_representation <= representation
analytical_representation_name	representation.name	43		analytical_representation <= representation representation.name
analytical_representation to analytical_model (as model_representation)	PATH			analytical_representation <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => analytical_model
analytical_representation to parameter_assignment (as model_parameter_assignment)	PATH			analytical_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 ->

Table 47 – Mapping table for functional_unit_analytical_representation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation => parameter_assignment_representation

Table 48 – Mapping table for functional_usage_view UoF

Application element	AIM element	Source	Rules	Reference path
EQUIVALENT_FUNCTIONAL_TERMINALS_ASSIGNMENT	shape_aspect	41	164	{shape_aspect shape_aspect.description = 'equivalent functional terminals'}
name	shape_aspect.name	41		
equivalent_functional_terminals_assignment to functional_usage_view - terminal_definition (as equivalent_terminals)	PATH		333	shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'equivalent functional terminals assignment'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => functional_unit_terminal_definition
FUNCTIONAL_TERMINAL_GROUP	functional_terminal_group	210		functional_terminal_group <= group
name	group.name	41		functional_terminal_group <= group group.name
group_description	group.description	41		functional_terminal_group <= group group.description
group_type	group.id	41		functional_terminal_group <= group group.id
FUNCTIONAL_TERMINAL_GROUP_ASSIGNMENT	applied_group_assignment	210		{applied_group_assignment applied_group_assignment <= group_assignment group_assignment.assigned_group ->

Table 48 – Mapping table for functional_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				group => functional_terminal_group}
functional_terminal_group_- assignment to functional_terminal_group (as composed_group)	PATH			applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => functional_terminal_group
functional_terminal_group_- assignment to scalar_terminal_definition (as functional_usage_view_terminal)	PATH			applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = shape_aspect {shape_aspect shape_aspect.description = 'scalar terminal'} shape_aspect => functional_unit_terminal_definition
FUNCTIONAL_UNIT_USAGE_- VIEW	(functional_unit) (externally_defined_functional_- unit) (library_defined_functional_unit)	210 210 210	338	(functional_unit <= product_definition) (externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) (library_defined_functional_unit <= externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional design usage'}
FUNCTIONAL_UNIT_USAGE_- VIEW_TERMINAL_DEFINI- TION	functional_unit_terminal_- definition	210	181	functional_unit_terminal_definition <= shape_aspect

Table 48 – Mapping table for functional_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
signal_name	shape_aspect.name	41	333	functional_unit_terminal_definition <= shape_aspect shape_aspect.name
functional_unit_usage_view_- terminal_definition to functional_unit_usage_view (as associated_functional_unit_- definition)	PATH		340, 338	shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'functional design usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
FUNCTIONAL_UNIT_USAGE_- VIEW_TERMINAL_DEFINI- TION_MAKE_FROM_RELA- TIONSHIP	make_from_functional_unit_- terminal_definition_relationship	210		{[make_from_functional_unit_terminal_- definition_relationship <= shape_aspect_relationship] [make_from_functional_unit_terminal_- definition_relationship <= shape_aspect]}

Table 48 – Mapping table for functional_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
functional_unit_usage_view_- terminal_definition_make_from_- relationship to functional_unit_usage_view_- terminal_definition (as reusable_feature)	PATH			make_from_functional_unit_terminal_- definition_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => functional_unit_terminal_definition
functional_unit_usage_view_- terminal_definition_make_from_- relationship to functional_unit_usage_view_- terminal_definition (as resultant_feature)	PATH			make_from_functional_unit_terminal_- definition_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => functional_unit_terminal_definition
functional_unit_usage_view_- terminal_definition_make_from_- relationship to engineering_make_from (as associated_make_from)	PATH			make_from_functional_unit_terminal_- definition_relationship <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => design_make_from_relationship
SCALAR_TERMINAL_DEFINI- TION	functional_unit_terminal_- definition	210		functional_unit_terminal_definition <= shape_aspect {shape_aspect shape_aspect.description = 'scalar terminal'}
SCALAR_TERMINAL_DEFINI- TION_LINK	scalar_terminal_definition_link	210		{[scalar_terminal_definition_link <= shape_aspect] [scalar_terminal_definition_link <= shape_aspect_relationship]}

Table 48 – Mapping table for functional_usage_view UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
scalar_terminal_definition_link to scalar_terminal_definition (as precedent_terminal_definition)	PATH			scalar_terminal_definition_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => {shape_aspect shape_aspect.description = 'scalar terminal'} functional_unit_terminal_definition
scalar_terminal_definition_link to scalar_terminal_definition (as subsequent_terminal_definition)	PATH			scalar_terminal_definition_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect shape_aspect.description = 'scalar terminal'} functional_unit_terminal_definition

Table 49 – Mapping table for functional_usage_view_terminal_to_analytical_model_port_mapping UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_MODEL_- PORT_ASSIGNMENT	property_definition_representation	41		
analytical_model_port_assignment to analytical_model_port (as assigned_analytical_model_port)	PATH		342	property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port
analytical_model_port_assignment to functional_unit_usage_view_- terminal_definition (as assigned_functional_unit_- terminal)	PATH			property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => functional_unit_terminal_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF

Application element	AIM element	Source	Rules	Reference path
ANGULAR_DIMENSION_- WITH_DIRECTION_VECTOR	angular_dimension_with_- orientation	210	335, 7	angular_dimension_with_orientation <= dimensional_location
angular_dimension_with_- direction_vector to axis_placement (as measurement_orientation)	PATH		335, 344, 342, 109	angular_dimension_with_orientation <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.description = 'dimensional location property'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement
angular_dimension_with_- direction_vector to datum_based_vector_orientation (as measurement_orientation)	PATH			angular_dimension_with_orientation <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition property_definition.description = 'dimensional location property'} property_definition <- property_definition.relationship.relatiing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'measurement orientation'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'datum based vector orientation'} </pre>
ANGULAR_SIZE_DIMENSION	angular_size	47	8	
half_angle	representation_item.name	43	336, 111, 342	<pre> angular_size <= dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_- representation.dimension dimensional_characteristic_representation dimensional_characteristic_- representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item representation_item.name {(representation_item.name = 'full angle') (representation_item.name = 'half angle')} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
major_angle	angular_size.angle_selection	47		{(angular_size.angle_selection = .LARGE.) (angular_size.angle_selection = .SMALL.) (angular_size.angle_selection = .EQUAL.)}
ANGULARITY_TOLERANCE	angularity_tolerance	210	352	angularity_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'angularity'}
APEX	apex	47		
AXIS_PLACEMENT	(axis2_placement_2d) (axis2_placement_3d)	42 42		(axis2_placement_2d) (axis2_placement_3d)
axis_placement to cartesian_point (as translation)	PATH			(axis2_placement_2d <= placement placement.location ->) (axis2_placement_3d <= placement placement.location ->) cartesian_point
axis_placement to direction (as reference_direction)	PATH			(axis2_placement_2d axis2_placement_2d.ref_direction ->) (axis2_placement_3d axis2_placement_3d.ref_direction ->) direction
AXIS_PLACEMENT_2D	axis2_placement_2d	42		
AXIS_PLACEMENT_3D	axis2_placement_3d	42		
BOUNDARY_BASED_POSI- TION_TOLERANCE	[position_tolerance] [modified_geometric_tolerance]	210 47	352	[position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [modified_geometric_tolerance <= geometric_tolerance]

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
boundary_based_position_- tolerance to profile_of_any_surface_- tolerance_with_referenced_- datum_system (as profile_control)	PATH		352	<pre> position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.relatiing_- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'profile_control'} geometric_tolerance_relationship.related_- geometric_tolerance -> {geometric_tolerance {geometric_tolerance.name = 'surface profile'} {geometric_tolerance.name = 'surface profile locating'} {geometric_tolerance.name = 'surface profile refinement'}} [geometric_tolerance => physical_unit_geometric_tolerance => surface_profile_tolerance] [geometric_tolerance => physical_unit_geometric_tolerance => geometric_tolerance_with_specified_datum_- system] </pre>
BOUNDARY_SIZE_CHARAC- TERISTIC	representation	43		<pre> {representation (representation.name = 'diametrical boundary size')} (representation.name = 'opposing boundary set size'))} </pre>
boundary_size_characteristic to length_data_element (as tolerance_value)	PATH		21, 342, 109	<pre> representation representation.items[i] -> {representation_item representation_item.name = 'tolerance value'} representation_item => measure_representation_item <= </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit => length_measure_with_unit
boundary_size_characteristic to length_data_element (as maximum_tolerance_value)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'maximum tolerance value'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
BOUNDARY_ZONE_DEFINITION_WITH_SPECIFIED_SIZE	property_definition	41	22, 333	{property_definition [property_definition.description = 'boundary zone definition with specified size'] [property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_definition = shape_aspect tolerance_zone_boundary]}}
boundary_zone_definition_with_- specified_size to boundary_size_characteristic (as specified_boundary_size)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation (representation.name = 'diametrical boundary size') (representation.name = 'opposing boundary set size')}}}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
CENTRE_AXIS	centre_of_symmetry	47	333	{centre_of_symmetry <= derived_shape_aspect <= shape_aspect shape_aspect.description = 'axis'}
CENTRE_OF_SYMMETRY	centre_of_symmetry	47		
CENTRE_PLANE	centre_of_symmetry	47	26, 333	{centre_of_symmetry <= derived_shape_aspect <= shape_aspect shape_aspect.description = 'plane'}
CENTRE_POINT	centre_of_symmetry	47	26	{centre_of_symmetry <= derived_shape_aspect <= shape_aspect shape_aspect.description = 'point'}
CIRCULAR_RUNOUT_TOLER- ANCE	circular_runout_tolerance	210	352	circular_runout_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'circular_runout'}
CIRCULARITY_TOLERANCE	physical_unit_geometric_tolerance	210		physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'circularity'}
COMMON_DATUM	physical_unit_datum	210	333	physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'common_datum'}
COMPOSITE_FEATURE_LO- CATING_POSITION_TOLER- ANCE	[position_tolerance] [geometric_tolerance_with_- specified_datum_system]	210 210	352	[position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <=

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_tolerance {[geometric_tolerance.name = 'feature locating position']} [geometric_tolerance <- geometric_tolerance_relationship.relatiing_- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship.name = 'position refining control'}] geometric_tolerance.related_- geometric_tolerance -> geometric_tolerance {geometric_tolerance.name = 'feature relating position'}] geometric_tolerance => physical_unit_geometric_tolerance => position_tolerance}} </pre>
COMPOSITE_FEATURE_RELATING_POSITION_TOLERANCE	position_tolerance	210		<pre> position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance.name = 'feature relating position'}} </pre>
composite_feature_relatiing_- position_tolerance to composite_feature_locating_- position_tolerance (as refined_control)	PATH		352	<pre> position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.related_- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship.name = 'position refining control'}] geometric_tolerance.related_- geometric_tolerance -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{geometric_tolerance geometric_tolerance.name = 'feature locating position'} geometric_tolerance => physical_unit_geometric_tolerance => [position_tolerance] [geometric_tolerance_with_specified_datum_- system]</pre>
<pre>composite_feature_relating_- position_tolerance to composite_feature_relating_- position_tolerance (as refined_control)</pre>	PATH			<pre>position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.related_- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'position refining_control'} geometric_tolerance_relationship.related_- geometric_tolerance -> {geometric_tolerance geometric_tolerance.name = 'feature relating position'} geometric_tolerance => physical_unit_geometric_tolerance => position_tolerance</pre>
COMPOSITE_GROUP_SHAPE_- ELEMENT	composite_group_shape_aspect	210	349	<pre>composite_group_shape_aspect <= composite_shape_aspect</pre>
COMPOSITE_LINEAR PRO- FILE_LOCATING_TOLERANCE	<pre>[[linear_profile_tolerance] [geometric_tolerance_with_- specified_datum_system]]</pre>	210 210	352	<pre>[[linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <=</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_tolerance {[geometric_tolerance.name = 'linear profile locating'] [geometric_tolerance <- geometric_tolerance_relationship.relati- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship.name = 'linear profile refining control'} geometric_tolerance_relationship.related_ geometric_tolerance -> geometric_tolerance {geometric_tolerance.name = 'linear profile refinement'} geometric_tolerance => physical_unit_geometric_tolerance => linear_profile_tolerance]} </pre>
COMPOSITE_LINEAR_PROFILE_REFINEMENT_TOLERANCE	linear_profile_tolerance	210		<pre> linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance.name = 'linear profile refinement'} </pre>
composite_linear_profile_refinement_tolerance to composite_linear_profile_locating_tolerance (as refined_control)	PATH		352	<pre> linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.related_ geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship.name = 'linear profile refining control'} geometric_tolerance_relationship.related_ geometric_tolerance -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{geometric_tolerance geometric_tolerance.name = 'linear profile locating'} geometric_tolerance => physical_unit_geometric_tolerance => [linear_profile_tolerance] [geometric_tolerance_with_specified_datum_- system]</pre>
composite_linear_profile_- refinement_tolerance to composite_linear_profile_- refinement_tolerance (as refined_control)	PATH			<pre>linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance.relationship.related_- geometric_tolerance geometric_tolerance.relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'linear profile refining control'} geometric_tolerance.relationship.related_- geometric_tolerance -> {geometric_tolerance geometric_tolerance.name = 'linear profile refinement'} geometric_tolerance => physical_unit_geometric_tolerance => linear_profile_tolerance</pre>
COMPOSITE_SHAPE_ELEMENT	composite_shape_aspect	47	55, 349, 333	<pre>{composite_shape_aspect <= shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composing'}}}</pre>
COMPOSITE_SURFACE_PROFILE_LOCATING_TOLERANCE	[surface_profile_tolerance] [geometric_tolerance_with_- specified_datum_system]	210 210	352	<pre>[surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <=</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_tolerance {[geometric_tolerance.name = 'surface profile locating'] [geometric_tolerance <- geometric_tolerance_relationship.relati- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'surface profile refining control'} geometric_tolerance_relationship.related_ geometric_tolerance -> geometric_tolerance {geometric_tolerance.name = 'surface profile refinement'} geometric_tolerance => physical_unit_geometric_tolerance => surface_profile_tolerance]} </pre>
COMPOSITE_SURFACE_PROFILE_REFINEMENT_TOLERANCE	surface_profile_tolerance	210		<pre> surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance.name = 'surface profile refinement'} </pre>
composite_surface_profile_refinement_tolerance to composite_surface_profile_locating_tolerance (as refined_control)	PATH		352	<pre> surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.related_ geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'surface profile refining control'} geometric_tolerance_relationship.related_ geometric_tolerance -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{geometric_tolerance geometric_tolerance.name = 'surface profile locating'} geometric_tolerance => physical_unit_geometric_tolerance => [surface_profile_tolerance] [geometric_tolerance_with_specified_datum_- system]</pre>
composite_surface_profile_- refinement_tolerance to composite_surface_profile_- refinement_tolerance (as refined_control)	PATH			<pre>surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance_relationship.related_- geometric_tolerance geometric_tolerance_relationship {geometric_tolerance_relationship geometric_tolerance_relationship.name = 'surface profile refining control'} geometric_tolerance_relationship.related_- geometric_tolerance -> geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'surface profile refinement'} geometric_tolerance => physical_unit_geometric_tolerance => surface_profile_tolerance</pre>
COMPOSITE_UNIT_SHAPE_- ELEMENT	composite_unit_shape_aspect	210	349	<pre>composite_unit_shape_aspect <= composite_shape_aspect</pre>
CONCENTRICITY_AND_- COAXIALITY_TOLERANCE	concentricity_tolerance	210	352	<pre>concentricity_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{geometric_tolerance geometric_tolerance.name = 'concentricity'}
CONICAL_TOLERANCE_- ZONE_BOUNDARY	tolerance_zone_boundary	210	333	tolerance_zone_boundary <= shape_aspect {shape_aspect shape_aspect.description = 'conical'}
CONICAL_TOLERANCE_- ZONE_BOUNDARY_AND_- SURFACE_RELATIONSHIP	shape_aspect_relationship	41	62	{shape_aspect_relationship shape_aspect_relationship.name = 'conical' tolerance zone boundary and surface'}
conical_tolerance_zone_- boundary_and_surface_- relationship to conical_tolerance_zone_boundary (as sized_tolerance_zone_boundary)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conical'}
conical_tolerance_zone_- boundary_and_surface_- relationship to physical_feature_or_part_template (as terminating_surface)	PATH			tolerance_zone_boundary shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {(shape_aspect shape_aspect.product_definitional = .TRUE.) (shape_aspect => part_template_definition)}
CONICAL_TOLERANCE_- ZONE_BOUNDARY_DEFINI- TION	property_definition	41	63	{property_definition property_definition.description = 'conical' tolerance zone boundary'}
conical_tolerance_zone_- boundary_definition to conical_tolerance_zone_- boundary_diametrical_size_- characteristic (as boundary_sizes)	PATH			property_definition <- property_definition_relationship.relatng_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'boundary_size'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'conical' tolerance zone boundary size'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
conical_tolerance_zone_- boundary_definition to conical_tolerance_zone_boundary (as defined_zone_boundary)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'conical'} shape_aspect => tolerance_zone_boundary
CONICAL_TOLERANCE_- ZONE_BOUNDARY_DIAMET- RICAL_SIZE_CHARACTERIS- TIC	property_definition	41	64	{property_definition property_definition.description = 'conical tolerance zone boundary size'}
conical_tolerance_zone_- boundary_diametrical_size_- characteristic to conical_tolerance_zone_- boundary_and_surface_- relationship (as sized_boundary_at_surface)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'conical tolerance zone boundary and surface'}
conical_tolerance_zone_- boundary_diametrical_size_- characteristic to length_data_element (as tolerance_value)	PATH		342, 109	property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'conical tolerance zone boundary value'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
CONICAL_TOLERANCE_- ZONE_DEFINITION	tolerance_zone_definition	47	65, 333	{tolerance_zone_definition tolerance_zone_definition.zone -> tolerance_zone <= shape_aspect {shape_aspect shape_aspect.description = 'conical'}}
conical_tolerance_zone_definition to conical_tolerance_zone_- boundary_definition (as boundary_definition)	PATH		333	tolerance_zone_definition tolerance_zone_definition.boundaries[i] -> {[shape_aspect => tolerance_zone_boundary] [shape_aspect.description = 'conical']} shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition property_definition.definition <- property_definition {property_definition property_definition.description = 'conical' tolerance_zone_boundary'}
CURVE_DIMENSION	curve_dimension	210	336	curve_dimension <= dimensional_size
CYLINDRICITY_TOLERANCE	physical_unit_geometric_tolerance	210	352	physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'cylindricity'}
DATUM	physical_unit_datum	210	333	physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_AXIS	physical_unit_datum	210		physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.description = 'axis'}
DATUM_AXIS_AND_TOLER- ANCE_ZONE_ORIENTING_RE- LATIONSHIP	shape_aspect_relationship	41	74	{shape_aspect_relationship shape_aspect_relationship.name = 'datum axis and tolerance zone'}
datum_axis_and_tolerance_zone - orienting_relationship to tolerance_zone (as orienting_tolerance_zone)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => tolerance_zone
datum_axis_and_tolerance_zone - orienting_relationship to datum_axis (as orienting_datum_axis)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'axis'}
DATUM_AXIS_RELATED_ORI- ENTATION	property_definition	41	75	{property_definition property_definition.description = 'datum axis related orientation'}
orientation_type	property_definition.name	41		property_definition property_definition.name {(property_definition.name = 'centred on arc') (property_definition.name = 'centred on radial')}
datum_axis_related_orientation to datum_axis_and_tolerance_zone - orienting_relationship (as orienting_relationship)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'datum axis and tolerance zone'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_BASED_DIRECTION_VECTOR	property_definition	41	76	property_definition property_definition.description {property_definition.description = 'datum direction property'}
positive_direction_vector	property_definition.name	41		property_definition property_definition.name {(property_definition.name = 'positive direction vector') (property_definition.name = 'negative direction vector')}
datum_based_direction_vector to axis_placement (as direction_vector_representation)	PATH		342, 109	property_definition <= property_definition_representation property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement => axis1_placement
datum_based_direction_vector to datum_plane (as vector_of)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'plane'} shape_aspect => physical_unit_datum
DATUM_BASED_VECTOR_ORIENTATION	property_definition	41	78	{[property_definition property_definition.description = 'datum based vector orientation'] [property_definition property_definition.definition ->

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition shape_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'vector'}}]]}
datum_based_vector_orientation to direction_element (as direction_components)	PATH			property_definition <- property_definition_relationship.relationship. property_definition {property_definition_relationship {property_definition_relationship.name = 'direction_component'}} property_definition_relationship.relationship.related_ property_definition -> {property_definition {property_definition.description = 'direction element'}}
DATUM_FEATURE	physical_unit_datum_feature	210	333	physical_unit_datum_feature <= shape_aspect
identification	property_definition.name	41	333	physical_unit_datum_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition {property_definition.description = 'datum feature identification'}} property_definition.name

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_FEATURE_MATERIAL_CONDITION_PROPERTY	property_definition	41	79	{property_definition property_definition.description = 'datum feature material condition property'}
name	property_definition.name	41		property_definition property_definition.name {(property_definition.name = 'least material requirement') (property_definition.name = 'maximum material principle')}
datum_feature_material_ - condition_property to datum_feature_usage_in_datum_ - system_relationship (as applied to)	PATH			property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship.name = 'datum feature usage in datum system'}
DATUM_FEATURE_USAGE_- IN_COMMON_DATUM_RELATIONSHIP	shape_aspect_relationship	41	80	{shape_aspect_relationship shape_aspect_relationship.name = 'datum feature usage in common datum'}
datum_feature_usage_in_- common_datum_relationship to common_datum (as relating_shape_element)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.name = 'common datum'} shape_aspect => physical_unit_datum
DATUM_FEATURE_USAGE_- IN_DATUM_RELATIONSHIP	shape_aspect_relationship	41	81, 82	{shape_aspect_relationship (shape_aspect_relationship.name = 'datum feature usage in common datum') (shape_aspect_relationship.name = 'datum feature usage in single datum')}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
datum_feature_usage_in_datum_- relationship to datum (as relating_shape_element)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => physical_unit_datum
datum_feature_usage_in_datum_- relationship to datum_feature (as related_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum_feature
DATUM_FEATURE_USAGE_- IN_DATUM_SYSTEM_RELA- TIONSHIP	shape_aspect_relationship	41	83	{shape_aspect_relationship shape_aspect_relationship.name = 'datum feature usage in datum system'}
datum_feature_usage_in_datum_- system_relationship to datum_system (as established_datum_system)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => datum_system
datum_feature_usage_in_datum_- system_relationship to datum_feature (as used_datum_feature)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum_feature
DATUM_FEATURE_USAGE_- IN_SINGLE_DATUM_RELA- TIONSHIP	shape_aspect_relationship	41	84	{shape_aspect_relationship shape_aspect_relationship.name = 'datum feature usage in single datum'}
datum_feature_usage_in_single_- datum_relationship to single_datum (as relating_shape_element)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_PLANE	physical_unit_datum	210		physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.description = 'plane'}
DATUM_POINT	physical_unit_datum	210	333	physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.description = 'point'}
datum_point to cartesian_point (as location_in_2d)	PATH			shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation.used_ - representation -> representation representation.items[i] -> representation_item => geometric_representation_item => cartesian_point
datum_point to cartesian_point (as location_in_3d)	PATH			physical_unit_datum <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation.used_ - representation -> representation representation.items[i] ->

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => geometric_representation_item => cartesian_point
datum_point to volume_shape_intersection (as associated_volume_shape_- intersection)	PATH			physical_unit_datum <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation.used_- representation -> representation => volume_shape_intersection
DATUM_PRECEDENCE_AS- SIGNMENT	property_definition	41	85, 86	{property_definition property_definition.description = 'datum precedence assignment'}
name	property_definition.name	47		property_definition property_definition.name {(property_definition.name = 'primary') (property_definition.name = 'secondary') (property_definition.name = 'tertiary')}}
datum_precedence_assignment to datum_usage_in_datum_system_- relationship (as assigned_to)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect_relationship shape_aspect_relationship.name = 'datum usage in datum system'}
DATUM_REFERENCE_FRAME	datum_reference_frame	210	333	datum_reference_frame <= shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}
datum_reference_frame to datum_plane (as reference_planes)	PATH		333	datum_reference_frame <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference plane'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'plane'} shape_aspect => physical_unit_datum
datum_reference_frame to datum_axis (as reference_axes)	PATH			datum_reference_frame <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference axis'} shape_aspect_relationship.related_shape_aspect -> {[shape_aspect => derived_shape_aspect] [shape_aspect shape_aspect.description = 'axis']} shape_aspect => physical_unit_datum

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
datum_reference_frame to datum_point (as reference_origin)	PATH		333	<pre> datum_reference_frame <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference origin'} shape_aspect_relationship.related_shape_aspect -> {[shape_aspect => derived_shape_aspect] [shape_aspect shape_aspect.description = 'point']} shape_aspect => physical_unit_datum </pre>
datum_reference_frame to perpendicular_shape_element_- relationship (as reference_plane_pairs)	PATH			<pre> datum_reference_frame <= shape_aspect characterized_definition = shape_aspect characterized_definition <- property_definition.definition property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'reference plane pair'} property_definition_relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition = shape_aspect_- relationship {shape_aspect_relationship shape_aspect_relationship.name = 'perpendicular'} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
datum_reference_frame to perpendicular_shape_element - relationship (as reference_axis_pairs)	PATH			datum_reference_frame <= shape_aspect characterized_definition = shape_aspect characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'reference_axis pair'} property_definition.relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition = shape_aspect_ relationship {shape_aspect_relationship shape_aspect_relationship.name = 'perpendicular'}
DATUM_SYSTEM	datum_system	210		datum_system <= shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}
datum_system to datum_reference_frame (as established_datum_reference_ frame)	PATH		333	datum_system <= shape_aspect <- shape_aspect.relationship.related_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'established datum reference frame'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.related_shape_aspect -> shape_aspect => datum_reference_frame
DATUM_SYSTEM_DEFINING_- CONDITIONS	property_definition	41	87	<pre>{property_definition (property_definition.description = 'datum system property with material conditions') (property_definition.description = 'datum system property without material conditions')}</pre>
datum_system_defining_- conditions to datum_precedence_assignment (as assigned_datum_precedences)	PATH			<pre>property_definition <- property_definition_relationship.relatin_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'assigned datum precedence'}</pre>
datum_system_defining_- conditions to datum_system (as conditions_of)	PATH		333	<pre>property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => datum_system</pre>
DATUM_SYSTEM_DEFINING_- CONDITIONS_WITH_MATE- RIAL_CONDITIONS	property_definition	41	88	<pre>{property_definition property_definition.description = 'datum system property with material conditions'}</pre>
datum_system_defining_- conditions_with_material_- conditions to datum_feature_material_- condition_property (as applied_material_condition_- properties)	PATH			<pre>property_definition <- property_definition_relationship.relatin_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'applied material condition property'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'datum feature material condition property'}</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DATUM_SYSTEM_DEFINING_- CONDITIONS_WITHOUT_- MATERIAL_CONDITIONS	property_definition	41	89	{property_definition property_definition.description = 'datum system property without material conditions'}
DATUM_TARGET	physical_unit_datum_target	210	333	physical_unit_datum_target <= shape_aspect
DATUM_TARGET_SET	physical_unit_datum_target_set	210	333	physical_unit_datum_target_set <= physical_unit_datum_feature <= shape_aspect
DATUM_TARGET_USAGE_IN_- DATUM_TARGET_SET_RELA- TIONSHIP	shape_aspect_relationship	41	90, 91	{shape_aspect_relationship shape_aspect_relationship.name = 'datum target usage'}
datum_target_usage_in_datum_- target_set_relationship to datum_target_set (as defined_datum_target_set)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_unit_datum_feature => physical_unit_datum_target_set
datum_target_usage_in_datum_- target_set_relationship to datum_target (as used_datum_target)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum_target
datum_target_usage_in_datum_- target_set_relationship to positive_integer (as datum_target_number)	shape_aspect_- relationship.description	41		
DATUM_USAGE_IN_DATUM_- SYSTEM_RELATIONSHIP	shape_aspect_relationship	41	92, 93	{shape_aspect_relationship shape_aspect_relationship.name = 'datum usage in datum system'}
datum_usage_in_datum_system_- relationship to datum_system (as established_datum_system)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => datum_system

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
datum_usage_in_datum_system_- relationship to datum (as used_datum)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_unit_datum
DERIVED_SHAPE_ELEMENT	derived_shape_aspect	47	113	
DIAMETRICAL_BOUNDARY_- SIZE_CHARACTERISTIC	representation	43	131	{representation representation.name = 'diametrical boundary size'}
diametrical_boundary_size_- characteristic to tolerance_zone_circular_or_- cylindrical_or_spherical_boundary (as sized_boundary)	PATH		22, 333	representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition {property_definition property_definition.description = 'boundary zone_definition with specified size'} property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'circular or cylindrical or spherical'} shape_aspect => tolerance_zone_boundary
DIMENSION_ALONG_CURVE	dimensional_location_with_path	47	132, 133	
dimension_along_curve to shape_element (as path)	PATH			dimensional_location_with_path dimensional_location_with_path.path -> shape_aspect

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DIMENSION_LOCATION_- WITH_DIRECTION_VECTOR	dimensional_location_with_ direction	210	335, 134	dimensional_location_with_direction <= dimensional_location
dimension_location_with_ direction_vector to axis_placement (as measurement_direction)	PATH		335, 344, 109	dimensional_location_with_direction <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.description = 'dimensional location property'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement
dimension_location_with_ direction_vector to datum_based_vector_orientation (as measurement_direction)	PATH			dimensional_location_with_direction <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <=

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition {property_definition property_definition.description = 'dimensional location property'} property_definition <- property_definition.relationship.relatiing_- property_definition property_definition.relationship {property_definition.relationship property_definition.relationship.name = 'measurement orientation'} property_definition.relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'datum based vector orientation'} </pre>
DIMENSION_LOCATION_- WITH_SPECIFIED_DATUM_- SYSTEM	datum_system_based_- dimensional_location	210	335	datum_system_based_dimensional_location <= dimensional_location
dimension_location_with_- specified_datum_system to datum_system_defining_- conditions_without_material_- conditions (as referenced_datum_system_- defining_conditions)	PATH		335, 344	<pre> datum_system_based_dimensional_location <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'dimensional location property'} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition <- property_definition_relationship.relatin_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'referenced datum system property'} property_definition_relationship.related_ property_definition -> property_definition {property_definition property_definition.description = 'datum system property without material conditions'} </pre>
DIMENSION_QUALIFIER	measure_qualification	45	135	<pre> {measure_qualification (measure_qualification.name = 'predefined dimension qualifier') (measure_qualification.name = 'user defined dimension qualifier')} </pre>
dimension_qualifier to dimensional_characteristic (as qualified_dimensions)	PATH		342	<pre> measure_qualification measure_qualification.qualified_measure -> measure_with_unit => measure_representation_item <= representation_item <- representation.items[i] representation => shape_representation => shape_dimension_representation <- dimensional_characteristic_ representation.representation dimensional_characteristic_representation dimensional_characteristic_ representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_ location) (dimensional_characteristic = dimensional_ size) </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DIMENSION_RELATED_POSITIONAL_BOUNDARY	positional_boundary	210	333	positional_boundary <= shape_aspect {shape_aspect.description = 'dimension related positional boundary'}
dimension_related_positional_- boundary to positional_boundary_member (as boundary_members)	PATH		333	positional_boundary <= shape_aspect <= shape_aspect_relationship.relying_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'boundary member'}
dimension_related_positional_- boundary to physical_feature_or_part_template (as constrained_physical_feature)	PATH			shape_aspect_relationship.relying_shape_aspect -> shape_aspect => positional_boundary_member positional_boundary <= shape_aspect <= shape_aspect_relationship.relying_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constrained feature'}
				shape_aspect_relationship.relying_shape_aspect -> shape_aspect {(shape_aspect shape_aspect.product_definitional = .TRUE.) (shape_aspect => part_template_definition))}
DIMENSION_RELATED_POSITIONAL_BOUNDARY_DEFINITION	property_definition	41	136	{property_definition property_definition.description = 'dimension related positional boundary property'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
dimension_related_positional_- boundary_definition to dimension_related_positional_- boundary (as defined_positional_boundary)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'dimension related positional boundary'} shape_aspect => positional_boundary
dimension_related_positional_- boundary_definition to positional_boundary_member_- definition (as characterized_boundary_- member_definitions)	PATH			property_definition <- property_definition.relationship.relatiing_- property_definition property_definition.relationship {property_definition_relationship property_definition_relationship.name = 'defined_boundary_member_property'} property_definition.relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'positional boundary member property'}
DIMENSION_SET	property_definition	41	137	{property_definition property_definition.description = 'dimension set'}
dimension_set to dimensional_characteristic (as associated_dimensions)	PATH		344	property_definition <- property_definition.relationship.relatiing_- property_definition property_definition.relationship {property_definition_relationship property_definition_relationship.name = 'set associated dimension'} property_definition.relationship.related_- property_definition ->

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{property_definition property_definition.description = 'dimensional location property'} property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location (property_definition => dimensional_size_property <= dimensional_size)</pre>
DIMENSIONAL_CHARACTER- ISTIC	(dimensional_location) (dimensional_size)	47 47	138	(dimensional_location) (dimensional_size)
dimension_type	(dimensional_size.name) (shape_aspect- relationship.description)	47 41	344	<pre>(dimensional_size dimensional_size.name {(dimensional_size.name = 'angular') (dimensional_size.name = 'arc length') (dimensional_size.name = 'controlled radius') (dimensional_size.name = 'diameter') (dimensional_size.name = 'linear') (dimensional_size.name = 'radius')}) (dimensional_location <= shape_aspect_relationship shape_aspect_relationship.description {(shape_aspect_relationship.description = 'angular') (shape_aspect_relationship.description = 'arc length')})</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect_relationship.description = 'controlled radius') (shape_aspect_relationship.description = 'diameter') (shape_aspect_relationship.description = 'linear') (shape_aspect_relationship.description = 'radius')})
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	(auxiliary_characteristic_dimension_representation) (dimensional_characteristic_representation)	210 47		(auxiliary_characteristic_dimension_representation <= dimensional_characteristic_representation) (dimensional_characteristic_representation)
auxiliary_dimension_representation	PATH			(auxiliary_characteristic_dimension_representation <= dimensional_characteristic_representation .TRUE.) (dimensional_characteristic_representation .FALSE.)
dimensional_characteristic_representation to dimensional_location (as dimension)	PATH			(auxiliary_characteristic_dimension_representation <= dimensional_characteristic_representation) (dimensional_characteristic_representation) dimensional_location -> dimensional_characteristic_dimensional_location dimensional_location (auxiliary_characteristic_dimension_representation <= dimensional_characteristic_representation) dimensional_location -> dimensional_characteristic_dimensional_location
dimensional_characteristic_representation to dimensional_size (as dimension)	PATH			(auxiliary_characteristic_dimension_representation <= dimensional_characteristic_representation) dimensional_size -> dimensional_characteristic_dimensional_size

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				dimensional_characteristic dimensional_characteristic = dimensional_size dimensional_size
dimensional_characteristic_- representation to shape_dimension_representation (as representation)	PATH			(auxiliary_characteristic_dimension_- representation <= dimensional_characteristic_representation) (dimensional_characteristic_representation) dimensional_characteristic_- representation.representation -> shape_dimension_representation
DIMENSIONAL_LOCATION	(directed_dimensional_location) (dimensional_location)	210 47	139, 335	(directed_dimensional_location <= dimensional_location) (dimensional_location)
directed	PATH			(directed_dimensional_location <= dimensional_location .TRUE.) (dimensional_location .FALSE.)
dimensional_location to shape_- element_locating_relationship (as located_elements)	IDENTICAL MAPPING			
DIMENSIONAL_SIZE	dimensional_size	47	140, 336	dimensional_size
envelope	representation.name	43	342, 111	dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_- representation.dimension dimensional_characteristic_representation dimensional_characteristic_- representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.name = 'envelope tolerance'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
dimensional_size_to_shape_element (as size_of)	PATH			dimensional_size dimensional_size.applies_to -> shape_aspect
DIMENSIONAL_SIZE_- BASED_ON_OPPOSING_- BOUNDARIES	opposing_boundary_- dimensional_size	210	336	opposing_boundary_dimensional_size <= dimensional_size
dimensional_size_based_on_- opposing_boundaries_to shape_element_locating_- relationship (as boundary_separation)	IDENTICAL MAPPING			
dimensional_size_based_on_- opposing_boundaries_to composite_unit_shape_element (as size_of)	PATH			opposing_boundary_dimensional_size <= dimensional_size dimensional_size.applies_to -> shape_aspect => composite_shape_aspect => composite_unit_shape_aspect
DIRECTED_AXIS	derived_shape_aspect	47	143, 144	{derived_shape_aspect <= shape_aspect shape_aspect.description = 'directed axis'}
directed_axis_to_centre_axis (as axis_to_be_oriented)	PATH		26, 333, 344	derived_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {[shape_aspect_relationship => shape_aspect_deriving_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'derived axis']}] shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'axis'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => derived_shape_aspect => centre_of_symmetry </pre>
directed_axis_to_shape_element (as end_shape_element)	PATH			<pre> derived_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {[shape_aspect_relationship => shape_aspect_deriving_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'end_shape aspect']} shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>
directed_axis_to_shape_element (as start_shape_element)	PATH		333, 344	<pre> derived_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {[shape_aspect_relationship => shape_aspect_deriving_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'start_shape aspect']} shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>
DIRECTION_ELEMENT	property_definition	41	145	<pre> {property_definition property_definition.description = 'direction element'} </pre>
direction_element_to angle_data_element (as angle_value)	PATH		109	<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> {representation_item representation_item.name = 'angle value' representation_item => measure_representation_item <= plane_angle_measure_with_unit } </pre>
direction_element to datum_based_direction_vector (as direction)	PATH			<pre> property_definition <- property_definition_relationship.relatin_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'element direction vector'} property_definition_relationship.related_ - property_definition -> property_definition {property_definition property_definition.description = 'datum direction property'} </pre>
direction_element to datum_based_direction_vector (as direction_of_half_datum_plane)	PATH			<pre> property_definition <- property_definition_relationship.relatin_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'half datum plane direction vector'} property_definition_relationship.related_ - property_definition -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition { property_definition property_definition.description = 'datum direction property'} </pre>
direction_element to datum_plane (as base_datum_plane)	PATH		333	<pre> property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship { shape_aspect_relationship shape_aspect_relationship.name = 'datum plane and half plane relationship'} [shape_aspect_relationship shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect { shape_aspect [shape_aspect.description = 'half plane'] [shape_aspect.product_definitional = .FALSE.]}] shape_aspect_relationship.related_shape_aspect -> shape_aspect => { shape_aspect shape_aspect.description = 'plane'} physical_unit_datum </pre>
EXTERNALLY_DEFINED_DIMENSION_REPRESENTATION	shape_dimension_representation	47		
externally_defined_dimension_ - representation to external_definition (as definitions)	PATH		342	<pre> shape_dimension_representation <= shape_representation <= representation document_assigned_item = representation </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint
externally_defined_dimension_- representation to data_element (as dimension_value)	PATH			shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit
FLATNESS_TOLERANCE	physical_unit_geometric_tolerance	210	352	physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'flatness'}
GEOMETRIC_ALIGNMENT	geometric_alignment	47		
GEOMETRIC_INTERSECTION	geometric_intersection	47		
GEOMETRIC_TOLERANCE	physical_unit_geometric_tolerance	210		[physical_unit_geometric_tolerance <= geometric_tolerance] [physical_unit_geometric_tolerance <= property_definition]
geometric_tolerance to conical_- tolerance_zone_boundary_- diametrical_size_characteristic (as tolerance_zone_or_boundary_- definition)	PATH		340	physical_unit_geometric_tolerance <= property_definition <- property_definition_relationship.relatiing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'defined boundary'} property_definition_relationship.related_- property_definition -> property_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition property_definition.description = 'conical tolerance zone boundary size'}
geometric_tolerance to positional_boundary_member_- definition (as tolerance_zone_or_boundary_- definition)	PATH			physical_unit_geometric_tolerance <= property_definition <- property_definition_relationship.relatiing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'defined boundary'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'positional boundary member property'}
geometric_tolerance to profile_related_positional_- boundary_definition (as tolerance_zone_or_boundary_- definition)	PATH		340	physical_unit_geometric_tolerance <= property_definition <- property_definition_relationship.relatiing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'defined boundary'} property_definition_relationship.related_- property_definition -> property_definition property_definition {property_definition property_definition.description = 'profile related positional boundary property'}
geometric_tolerance to tolerance_zone_definition (as tolerance_zone_or_boundary_- definition)	PATH		352	physical_unit_geometric_tolerance <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
geometric_tolerance to centre_axis (as tolerated_shape_element)	PATH		26, 55, 352, 333	<pre> physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({shape_aspect shape_aspect.description = 'axis'}) shape_aspect => derived_shape_aspect => centre_of_symmetry) ({shape_aspect => composite_shape_aspect}) shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'composing'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'axis'} shape_aspect => derived_shape_aspect => centre_of_symmetry) </pre>
geometric_tolerance to centre_plane (as tolerated_shape_element)	PATH		26, 55	<pre> physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({shape_aspect shape_aspect.description = 'plane'}) shape_aspect => derived_shape_aspect => centre_of_symmetry) ({shape_aspect => composite_shape_aspect}) shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship { shape_aspect_relationship.name = 'composing' } shape_aspect_relationship.related_shape_aspect -> { shape_aspect shape_aspect.description = 'plane' } shape_aspect => derived_shape_aspect => centre_of_symmetry </pre>
geometric_tolerance to centre_point (as toleranced_shape_element)	PATH		26, 55, 352, 333	<pre> physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({ shape_aspect shape_aspect.description = 'point' }) shape_aspect => derived_shape_aspect => centre_of_symmetry ({ shape_aspect => composite_shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship { shape_aspect_relationship.name = 'composing' } shape_aspect_relationship.related_shape_aspect -> { shape_aspect shape_aspect.description = 'point' } shape_aspect => derived_shape_aspect => centre_of_symmetry }) physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({ shape_aspect </pre>
geometric_tolerance to tangent_plane (as toleranced_shape_element)	PATH		55	<pre> physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({ shape_aspect </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.description = 'plane' } shape_aspect => derived_shape_aspect => tangent) ({shape_aspect => composite_shape_aspect} shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'composing'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'plane'} shape_aspect => derived_shape_aspect => tangent) </pre>
geometric_tolerance to physical_feature_or_part_template (as toleranced_shape_element)	PATH		55, 352, 333	<pre> physical_unit_geometric_tolerance <= geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> ({shape_aspect shape_aspect.product_definitional = .TRUE.}) part_template_definition)) shape_aspect) ({shape_aspect => composite_shape_aspect} shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'composing'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => part_template_definition)
GEOMETRIC_TOLERANCE_- QUALIFIER	measure_qualification	45	186	{measure_qualification (measure_qualification.name = 'predefined geometric_tolerance_qualifier') (measure_qualification.name = 'user defined geometric_tolerance_qualifier'))}
geometric_tolerance_qualifier to geometric_tolerance (as qualified_geometric_tolerances)	PATH		352	measure_qualification measure_qualification.qualified_measure -> measure_with_unit <- geometric_tolerance.magnitude geometric_tolerance => physical_unit_geometric_tolerance
GEOMETRIC_TOLERANCE_- WITH_APPLIED_MATERIAL_- CONDITION_PROPERTY	[modified_geometric_tolerance] [physical_unit_geometric_- tolerance]	47 210	187	{[modified_geometric_tolerance <= geometric_tolerance] [physical_unit_geometric_tolerance <= geometric_tolerance]}
applied_material_condition_prop- erty_name	modified_geometric_- tolerance.modifier	47		{(modified_geometric_tolerance.modifier = .MAXIMUM_MATERIAL_CONDITION.) (modified_geometric_tolerance.modifier = .LEAST_MATERIAL_CONDITION.)}
GEOMETRIC_TOLERANCE_- WITH_REFERENCED_DA- TUM_SYSTEM	geometric_tolerance_with_- specified_datum_system	210	352	geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance
geometric_tolerance_with_- referenced_datum_system to datum_system_defining_- conditions (as datum_system_definition)	PATH		340	geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= property_definition <- property_definition_relationship.relationship_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'referenced_datum_system_definition'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_relationship.related_ property_definition -> property_definition {property_definition (property_definition.description = 'datum system property with material conditions') (property_definition.description = 'datum system property without material conditions')}} </pre>
GEOMETRIC_TOLERANCE_ WITHOUT_REFERENCED_DA- TUM_SYSTEM	physical_unit_geometric_tolerance	210		<pre> [physical_unit_geometric_tolerance <= geometric_tolerance] [physical_unit_geometric_tolerance <= property_definition] </pre>
INTER_STRATUM_FEATURE	inter_stratum_feature	210	340, 338, 268	<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {[(shape_aspect => component_shape_aspect => inter_stratum_feature => (cutout_edge_segment) (interconnect_module_edge_segment) (plated_inter_stratum_feature)) (shape_aspect (shape_aspect.description = 'partially plated cutout') (shape_aspect.description = 'partially plated interconnect module edge') (shape_aspect.description = 'unsupported passage') (shape_aspect.description = 'cutout') (shape_aspect.description = 'physical connectivity interrupting cutout') (shape_aspect.description = 'dielectric material passage') (shape_aspect.description = 'interconnect module edge'))]} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition product_definition.description = 'laminare component'] [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence']] </pre>
feature_of_size	descriptive_representation_- item.description	45	333	<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation representation.items[i] -> {representation_item representation_item.name = 'feature of size'} representation_item => descriptive_representation_item {descriptive_representation_item.description (descriptive_representation_item.description = 'true')} (descriptive_representation_item.description = 'false')}} </pre>
<pre> inter_stratum_feature to inter_stratum_feature_template (as usage_definition) </pre>	PATH		333	<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'via template') (shape_aspect.description = 'component termination passage template') (shape_aspect.description = 'unsupported passage template') (shape_aspect.description = 'inter stratum feature edge template') (shape_aspect.description = 'inter stratum feature template')} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'counterbore passage template') (shape_aspect.description = 'countersunk passage template') (shape_aspect.description = 'blind passage template') } shape_aspect => part_template_definition</pre>
LIMIT_DIMENSION_REPRESENTATION	shape_dimension_representation	47	204	
limit_dimension_representation to data_element (as lower_dimension_value)	PATH		342	<pre>shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit {measure_with_unit <- measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'}} measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'lower value'}}</pre>
limit_dimension_representation to data_element (as upper_dimension_value)	PATH			<pre>shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item =></pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> measure_representation_item <= measure_with_unit <- {measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'}} measure_qualification.qualified[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'upper value'}} </pre>
LINEAR_EXTENSION	extension	47		
MAXIMUM_DIMENSION_RE- PRESENTATION	shape_dimension_representation	47	342	<pre> {shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit <- measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'}} measure_qualification.qualified[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'maximum dimension'}}} </pre>
MINIMUM_DIMENSION_RE- PRESENTATION	shape_dimension_representation	47		<pre> {shape_dimension_representation <= shape_representation <= representation </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit <- measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'} measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'minimum dimension'}}} </pre>
NON_FEATURE_SHAPE_ELEMENT	shape_aspect	41		<pre> {shape_aspect shape_aspect.product_definitional = .FALSE.} </pre>
non_feature_shape_element to footprint_definition (as scope)	PATH			<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => footprint_definition </pre>
non_feature_shape_element to padstack_definition (as scope)	PATH			<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => padstack_definition
non_feature_shape_element to physical_unit (as scope)	PATH			shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit
non_feature_shape_element to stratum (as scope)	PATH			shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
non_feature_shape_element to template_definition (as scope)	PATH			<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'} </pre>
NON_FEATURE_SHAPE_ELEMENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship.name = 'non feature shape element relationship'}
non_feature_shape_element_- relationship to non_feature_shape_element (as related_shape_element)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>
non_feature_shape_element_- relationship to non_feature_shape_element (as relating_shape_element)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect </pre>
OPPOSING_BOUNDARY_- SET_SIZE_CHARACTERISTIC	representation	43	218	{representation representation.name = 'opposing boundary set size'}
opposing_boundary_set_size_- characteristic to tolerance_zone_opposing_- boundary_set (as sized_boundary_set)	PATH		22, 342, 333	<pre> representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition {property_definition property_definition.description = 'boundary zone definition with specified size'} property_definition.definition -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition shape_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect shape_aspect => tolerance_zone_boundary (tolerance_zone_boundary => tolerance_zone_explicit_opposing_boundary_set) (tolerance_zone_boundary => tolerance_zone_implicit_opposing_boundary_set)
PARALLEL_SHAPE_ELEMENT_RELATIONSHIP	shape_aspect_relationship	41	226	{shape_aspect_relationship shape_aspect_relationship.name = 'parallel'}
PARALLELISM_TOLERANCE	parallelism_tolerance	210	352	parallelism_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'parallelism'}
PART_FEATURE	(shape_aspect) (composite_shape_aspect) (primary_orientation_feature) (secondary_orientation_feature) (package_body) (part_tooling_feature) (thermal_feature) (part_mounting_feature) (package_terminal) (assembly_module_terminal) (bare_die_terminal) (interconnect_module_terminal) (minimally_defined_bare_die_- terminal) (packaged_part_terminal) (package_body_surface)	41 210 210 210 210 210 210 210 210 210 210 210 210 210	232, 233	(composite_shape_aspect <= shape_aspect shape_aspect.description = 'part group feature') (shape_aspect (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')) (primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect) (secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect) (package_body <= shape_aspect)

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect (thermal_feature <= shape_aspect) (part_mounting_feature <= shape_aspect) (package_terminal <= shape_aspect) (assembly_module_terminal <= shape_aspect) (interconnect_module_terminal <= shape_aspect) (bare_die_terminal <= minimally_defined_bare_die_terminal <= shape_aspect) (minimally_defined_bare_die_terminal <= shape_aspect) (packaged_part_terminal <= shape_aspect) (package_body_surface <= shape_aspect)
material_state_change	PATH			shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface')}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect =>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {[representation representation.name = 'material state change']} [representation representation.items[i] -> representation_item {(representation_item.name = 'material addition') (representation_item.name = 'material removal')}}] shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature')}</pre>
name	PATH	41		

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect =>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_aspect.name </pre>
part_feature to part_feature (as precedent_feature)	PATH			<pre> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal {!{package_terminal => primary_reference_terminal}}) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_aspect <- </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'precedent feature'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
part_feature to physical_unit_usage_view (as associated_definition)	PATH		340, 338	shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'])}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect =>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)} shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(physical_unit => externally_defined_physical_unit => library_defined_physical_unit)}
PERPENDICULAR_SHAPE_- ELEMENT_RELATIONSHIP	shape_aspect_relationship	41	237	{shape_aspect_relationship shape_aspect_relationship.name = 'perpendicular'}
PERPENDICULARITY_TOLER- ANCE	perpendicularity_tolerance	210	352	perpendicularity_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'perpendicularity'}
PHYSICAL_COMPONENT_- FEATURE	(component_terminal) (component_interface_terminal) (component_feature) (component_mounting_feature)	210 210 210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'assembly module component_terminal') (shape_aspect.description = 'bare die component_terminal') (shape_aspect.description = 'interconnect component_join_terminal') (shape_aspect.description = 'interconnect module component_terminal') (shape_aspect.description = 'interconnect module component_stratum_based_terminal') (shape_aspect.description = 'minimally defined component_terminal') (shape_aspect.description = 'packaged component_join_terminal')}) (component_interface_terminal <= shape_aspect) (component_feature <= shape_aspect) (component_mounting_feature <= shape_aspect)

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_FEATURE_OR_PART_TEMPLATE	(part_template_definition) (shape_aspect)	210 41	340, 333	(part_template_definition <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'template definition'}) (shape_aspect {shape_aspect shape_aspect.product_definitional = .TRUE.})
PHYSICAL_LAMINATE_COMPONENT	(component_shape_aspect)	210 210	268, 333	component_shape_aspect <= shape_aspect {shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (material_removal_component_shape_aspect) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield)}} shape_aspect {shape_aspect

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'laminated text string component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component')}</pre>
PLUS_MINUS_TOLERANCE_- CHARACTERISTIC	coordinated_representation_item	210	109	<pre>coordinated_representation_item <= [representation] [representation_item {representation_item representation_item.name = 'plus minus tolerance'}]</pre>
plus_minus_tolerance_- characteristic to numeric_parameter (as component_characteristic)	PATH		109	<pre>coordinated_representation_item <= representation representation.items[i] -> representation_item => measure_representation_item</pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PLUS_MINUS_TOLERANCE_- RANGE_REPRESENTATION	shape_dimension_representation	47	112, 250	shape_dimension_representation <= shape_representation <= representation {representation.name = 'plus minus tolerance range representation'}
plus_minus_tolerance_range_- representation to data_element (as basic_dimension_value)	PATH			shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit {measure_with_unit <= measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'}}
plus_minus_tolerance_range_- representation to data_element (as upper_deviation)	PATH			measure_qualification.qualified_measure value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'basic value'}} shape_dimension_representation <= dimensional_characteristic_- representation.representation dimensional_characteristic_representation dimensional_characteristic_- representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_- location

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				dimensional_location dimensional_characteristic = dimensional_location (dimensional_characteristic = dimensional_size dimensional_size dimensional_characteristic = dimensional_size) dimensional_characteristic <- plus_minus_tolerance.toleranced_dimension. plus_minus_tolerance plus_minus_tolerance.range -> tolerance_method_definition tolerance_method_definition = tolerance_value tolerance_value tolerance_value.upper_bound -> measure_with_unit
plus_minus_tolerance_range_- representation to data_element (as lower_deviation)	PATH			shape_dimension_representation <- dimensional_characteristic_- representation.representation dimensional_characteristic_representation dimensional_characteristic_- representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_location dimensional_location dimensional_characteristic = dimensional_location) (dimensional_characteristic = dimensional_size dimensional_size dimensional_characteristic = dimensional_size) dimensional_characteristic <- plus_minus_tolerance.toleranced_dimension. plus_minus_tolerance plus_minus_tolerance.range ->

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				tolerance_method_definition tolerance_method_definition = tolerance_value tolerance_value tolerance_value.lower_bound -> measure_with_unit
POSITION_TOLERANCE	position_tolerance	210	352	position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance.name = 'position'} (geometric_tolerance.name = 'boundary' based position') (geometric_tolerance.name = 'feature locating position') (geometric_tolerance.name = 'feature relating position')}
POSITION_TOLERANCE_- WITH_REFERENCED_DA- TUM_SYSTEM	[position_tolerance] [geometric_tolerance_with_- specified_datum_system]	210 210		[position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance] {geometric_tolerance (geometric_tolerance.name = 'position') (geometric_tolerance.name = 'boundary' based position') (geometric_tolerance.name = 'feature locating position') (geometric_tolerance.name = 'feature relating position')}]
POSITION_TOLERANCE_- WITHOUT_REFERENCED_DA- TUM_SYSTEM	position_tolerance	210	352	position_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance (geometric_tolerance.name = 'position') (geometric_tolerance.name = 'feature relating position')}]

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
POSITIONAL_BOUNDARY	positional_boundary	210	333	positional_boundary <= shape_aspect {[shape_aspect (shape_aspect.description = 'dimension related positional boundary') (shape_aspect.description = 'profile related positional boundary')] [shape_aspect shape_aspect.product_definitional = .FALSE.]}
POSITIONAL_BOUNDARY_- AND_PROFILE_BOUNDARY_- MEMBER_RELATIONSHIP	shape_aspect_relationship	41	252	{shape_aspect_relationship shape_aspect_relationship.name = 'positional boundary and profile boundary member'}
positional_boundary_and_- profile_boundary_member_- relationship to profile_related_positional_- boundary (as offset_positional_boundary)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'profile related positional boundary'} shape_aspect => positional_boundary
positional_boundary_and_- profile_boundary_member_- relationship to tolerance_zone_boundary_- member (as base_profile_tolerance_zone_- boundary_member)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect [shape_aspect.description = 'tolerance zone boundary member'] [shape_aspect.product_definitional = .FALSE.]}
POSITIONAL_BOUNDARY_- DEFINITION	property_definition	41		{property_definition (property_definition.description = 'dimension related positional boundary property') (property_definition.description = 'profile related positional boundary property')}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
positional_boundary_definition to positional_boundary (as defined_positional_boundary)	PATH		333	{property_definition {property_definition.description = 'dimension related positional boundary property'} property_definition.definition -> characterized_definition shape_definition = shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'dimension related positional boundary'} shape_aspect => positional_boundary) (property_definition {property_definition.description = 'profile related positional boundary property'} property_definition.definition -> characterized_definition shape_definition = shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'profile related positional boundary'} shape_aspect => positional_boundary)
POSITIONAL_BOUNDARY_- MEMBER	positional_boundary_member	210		positional_boundary_member <= shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}
POSITIONAL_BOUNDARY_- MEMBER_DEFINITION	property_definition	41	253	{property_definition property_definition.description = 'positional boundary member property'}
positional_boundary_member_- definition to positional_boundary_member (as defined_positional_boundary_- member)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => positional_boundary_member

Table 50 – Mapping table for geometric_dimensioning_and_tolerance_UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
positional_boundary_member_- definition to length_data_element (as tolerance_value)	PATH		342, 109	property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'positional boundary member value'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
positional_boundary_member_- definition to dimensional_size (as associated_dimension)	PATH		336	property_definition <- property_definition_relationship.relatiing_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'boundary member associated dimension'} property_definition_relationship.related_ property_definition -> property_definition => dimensional_size_property <= dimensional_size
POSITIONAL_BOUNDARY_- OFFSET	property_definition	41	254	{property_definition property_definition.description = 'positional boundary offset'} property_definition
positional_boundary_offset to positional_boundary_and_- profile_boundary_member_- relationship (as offset_boundary_relationship)	PATH			property_definition property_definition.definition -> characterized_definition = shape_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'positional boundary and profile boundary member'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
positional_boundary_offset to length_data_element (as tolerance_value)	PATH		109	<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'positional boundary offset value'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
PRE_DEFINED_DIMENSION_- QUALIFIER	measure_qualification	45	255, 342, 111	<pre> {[measure_qualification measure_qualification.name = 'predefined dimension qualifier'] [measure_qualification measure_qualification.qualified_measure -> measure_with_unit => measure_representation_item <= representation_item <- representation.items[i] representation => shape_representation => shape_dimension_representation <- dimensional_characteristic_ representation.representation dimensional_characteristic_representation dimensional_characteristic_ representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_ location) (dimensional_characteristic = dimensional_ size)]]} </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
description	type_qualifier.name	45		measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name {(type_qualifier.name = 'average') (type_qualifier.name = 'free state')}
PRE_DEFINED_GEOMETRIC_- TOLERANCE_QUALIFIER	measure_qualification	45	256, 352	{[measure_qualification measure_qualification.name = 'predefined geometric_tolerance_qualifier'] [measure_qualification measure_qualification.qualified_measure -> measure_with_unit <- geometric_tolerance.magnitude geometric_tolerance => physical_unit_geometric_tolerance]}
description	type_qualifier.name	45		measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name {(type_qualifier.name = 'each element') (type_qualifier.name = 'each radial element') (type_qualifier.name = 'free state') (type_qualifier.name = 'major diameter') (type_qualifier.name = 'minor diameter') (type_qualifier.name = 'not concave') (type_qualifier.name = 'not convex') (type_qualifier.name = 'pitch diameter')}
PROFILE_BOUNDARY_DEFI- NITION	property_definition	41	269	{property_definition [property_definition.description = 'boundary zone definition with specified size'] [(property_definition.name = 'profile boundary') (property_definition.name = 'profile boundary with offsets')]}]

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
profile_boundary_definition to true_profile (as basic_profile)	PATH			property_definition <- property_definition.relationship.relatiing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'basic profile'}
profile_boundary_definition to tolerance_zone_opposing_- boundary_set (as defined_zone_boundary)	PATH		333	property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => tolerance_zone_boundary => (tolerance_zone_explicit_opposing_boundary_- set) (tolerance_zone_implicit_opposing_boundary_- set)
PROFILE_BOUNDARY_DEFINITION_WITH_OFFSETS	property_definition	41	270	{property_definition [property_definition.description = 'boundary zone definition with specified size'] [property_definition.name = 'profile boundary with offsets']}
profile_boundary_definition_- with_offsets to tolerance_zone_explicit_- opposing_boundary_set (as defined_zone_boundary)	PATH		22, 333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => tolerance_zone_boundary => tolerance_zone_explicit_opposing_boundary_set

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
profile_boundary_definition_- with_offsets to tolerance_zone_boundary_- member_offset (as boundary_offsets)	PATH			property_definition <- property_definition_relationship.relationship. property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'profile_tolerance zone boundary member offset'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'tolerance zone boundary member offset'}
PROFILE_OF_ANY_LINE_TOL- ERANCE	linear_profile_tolerance	210	352	linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance (geometric_tolerance.name = 'linear profile') (geometric_tolerance.name = 'linear profile locating') (geometric_tolerance.name = 'linear profile refinement')}
PROFILE_OF_ANY_LINE_- TOLERANCE_WITH_REFER- ENCED_DATUM_SYSTEM	[linear_profile_tolerance] [geometric_tolerance_with_- specified_datum_system]	210 210		[linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance] {geometric_tolerance (geometric_tolerance.name = 'linear profile') (geometric_tolerance.name = 'linear profile locating') (geometric_tolerance.name = 'linear profile refinement')}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PROFILE OF ANY LINE - TOLERANCE WITHOUT REF- ERENCED_DATUM_SYSTEM	linear_profile_tolerance	210	352	linear_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance (geometric_tolerance.name = 'linear profile') (geometric_tolerance.name = 'linear profile refinement')}}
PROFILE OF ANY SUR- FACE_TOLERANCE	surface_profile_tolerance	210	352	surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance (geometric_tolerance.name = 'surface profile') (geometric_tolerance.name = 'surface profile locating') (geometric_tolerance.name = 'surface profile refinement')}}
PROFILE OF ANY SUR- FACE_TOLERANCE_WITH_ REFERENCED_DATUM_SYS- TEM	[surface_profile_tolerance] [geometric_tolerance_with_ specified_datum_system]	210 210		[surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance] [geometric_tolerance_with_specified_datum_ system <= physical_unit_geometric_tolerance <= geometric_tolerance] {geometric_tolerance (geometric_tolerance.name = 'surface profile') (geometric_tolerance.name = 'surface profile locating') (geometric_tolerance.name = 'surface profile refinement')}]

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PROFILE_OF_ANY_SURFACE_TOLERANCE_WITHOUT_REFERENCED_DATUM_SYSTEM	surface_profile_tolerance	210	352	surface_profile_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance (geometric_tolerance.name = 'surface profile') (geometric_tolerance.name = 'surface profile refinement')}
PROFILE_RELATED_POSITIONAL_BOUNDARY	positional_boundary	210	333	positional_boundary <= shape_aspect {shape_aspect shape_aspect.description = 'profile related positional boundary'}
PROFILE_RELATED_POSITIONAL_BOUNDARY_DEFINITION	property_definition	41	271	{property_definition property_definition.description = 'profile related positional boundary property'}
profile_related_positional_boundary_definition to profile_related_positional_boundary (as defined_positional_boundary)	PATH		333	property_definition -> property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'profile related positional boundary'}
profile_related_positional_boundary_definition to positional_boundary_offset (as boundary_offset)	PATH			shape_aspect => positional_boundary property_definition <- property_definition.relationship.relationship.relat -> property_definition property_definition.relationship {property_definition_relationship property_definition_relationship.name = 'boundary offset'} property_definition -> property_definition property_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition property_definition.description = 'positional boundary offset'}
PROJECTED_TOLERANCE_- ZONE_DEFINITION	projected_zone_definition	47	272	
projected_tolerance_zone_- definition to projected_zone_height_- characteristic (as projected_zone_height)	IDENTICAL MAPPING			
PROJECTED_ZONE_AND_- BASE_RELATIONSHIP	projected_zone_definition	47	273	
projected_zone_and_base_- relationship to physical_feature_or_part_template (as base_physical_feature)	PATH		333	projected_zone_definition projected_zone_definition.projection_end -> shape_aspect {(shape_aspect shape_aspect.product_definition = .TRUE.) (shape_aspect => part_template_definition)}
projected_zone_and_base_- relationship to tolerance_zone (as projected_zone)	PATH			projected_zone_definition <= tolerance_zone_definition tolerance_zone_definition.zone -> tolerance_zone
PROJECTED_ZONE_HEIGHT_- CHARACTERISTIC	projected_zone_definition	47	274	
projected_zone_height_- characteristic to projected_zone_and_base_- relationship (as defined_height_of)	IDENTICAL MAPPING			

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
projected_zone_height_- characteristic to length_data_element (as projected_height_value)	PATH			projected_zone_definition projected_zone_definition.projected_length -> measure_with_unit => length_measure_with_unit
RESTRAINT_CONDITION	property_definition	41	286, 340	{property_definition (property_definition.description = 'restraint') (property_definition.description = 'tolerance specific restraint')}
restraint_condition to datum_feature_usage_in_datum_- relationship (as applied_to)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship (shape_aspect_relationship.name = 'datum feature usage in common datum') (shape_aspect_relationship.name = 'datum feature usage in single datum')}
restraint_condition to physical_feature_or_part_template (as applied_to)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {(shape_aspect shape_aspect.product_definitional = .TRUE.) (shape_aspect => part_template_definition)}
restraint_condition to ee_requirement_occurrence (as restraint_description)	PATH		340	property_definition <- property_definition_relationship.relaing_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'restraint description'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_relationship.related_ property_definition -> property_definition => requirements_property
RUNOUT_TOLERANCE_- ZONE_DEFINITION_WITH_- SPECIFIED_ANGLE	runout_zone_definition	47	293	runout_zone_definition
runout_tolerance_zone_- definition_with_specified_angle to runout_zone_orientation (as runout_orientation)	PATH		355	runout_zone_definition runout_zone_definition.orientation -> runout_zone_orientation => runout_zone_orientation_reference_direction
RUNOUT_TOLERANCE_- ZONE_ORIENTING_RELA- TIONSHIP	shape_aspect_relationship	41	294	{shape_aspect_relationship shape_aspect_relationship.name = 'runout tolerance zone orienting'}
runout_tolerance_zone_- orienting_relationship to tolerance_zone (as oriented_zone)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => tolerance_zone
runout_tolerance_zone_- orienting_relationship to directed_axis (as orienting_axis)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'directed axis'} shape_aspect => derived_shape_aspect
RUNOUT_ZONE_ORIENTA- TION	runout_zone_orientation_- reference_direction	47	295	
runout_zone_orientation to runout_tolerance_zone_- orienting_relationship (as oriented_relationship)	PATH			runout_zone_orientation_reference_direction runout_zone_orientation_reference_- direction.orienting_defining_relationship -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'runout tolerance zone orienting'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
runout_zone_orientation_to_angle_data_element (as semi_angle_value)	PATH		355	runout_zone_orientation_reference_direction <= runout_zone_orientation runout_zone_orientation.angle -> measure_with_unit => plane_angle_measure_with_unit
SEPARATE_GEOMETRIC_TOLERANCE_REQUIREMENT	geometric_tolerance_group	210	340	geometric_tolerance_group <= property_definition {property_definition property_definition.description = 'separate requirement'}
separate_geometric_tolerance_requirement_to_geometric_tolerance_with_referenced_datum_system (as separated_geometric_tolerances)	PATH		340	geometric_tolerance_group <= property_definition <- property_definition_relationship.relatiing_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'group geometric_tolerance'} property_definition_relationship.related_ property_definition -> property_definition => physical_unit_geometric_tolerance => geometric_tolerance_with_specified_datum_ system
SHAPE_DIMENSION_REPRESENTATION	shape_dimension_representation	47	297	
SHAPE_ELEMENT	shape_aspect	41	307	
description	descriptive_representation_item.description	45	342, 109	shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.description = 'shape element characterization' property_definition <- property_definition_representation property_definition_representation representation -> representation {representation representation.name = 'shape element description' representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description (shape_aspect {(shape_aspect.name = 'single datum') (shape_aspect.name = 'common datum') (shape_aspect.name = 'inter stratum join') (shape_aspect.name = 'intra stratum join') (shape_aspect.name = 'constrained intra stratum join')} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element characterization' property_definition <- property_definition_representation property_definition_representation property_definition_representation.used_ representation -> </pre>
name	<pre> (descriptive_representation_ item.description) (shape_aspect.name) </pre>	45 41		<pre> (shape_aspect {(shape_aspect.name = 'single datum') (shape_aspect.name = 'common datum') (shape_aspect.name = 'inter stratum join') (shape_aspect.name = 'intra stratum join') (shape_aspect.name = 'constrained intra stratum join')} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element characterization' property_definition <- property_definition_representation property_definition_representation property_definition_representation.used_ representation -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.name = 'shape element name' } representation.items[i] -> representation_item => descriptive_representation_item (shape_aspect (shape_aspect.name)) </pre>
SHAPE_ELEMENT_BOUNDARY_RELATIONSHIP	shape_aspect_relationship	41	298	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'boundary'} </pre>
SHAPE_ELEMENT_COMPOSING_RELATIONSHIP	shape_aspect_relationship	41	299	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'composing'} </pre>
shape_element_composing_relationship to composite_shape_element (as relating_shape_element)	PATH		333, 349	<pre> shape_aspect_relationship aspect -> shape_aspect => composite_shape_aspect </pre>
SHAPE_ELEMENT_CONSTITUENT_RELATIONSHIP	shape_aspect_relationship	41	300, 301	<pre> {shape_aspect_relationship (shape_aspect_relationship.name = 'bare die terminal surface constituent relationship') (shape_aspect_relationship.name = 'composing') (shape_aspect_relationship.name = 'constituent') (shape_aspect_relationship.name = 'interconnect module terminal surface constituent relationship') (shape_aspect_relationship.name = 'package terminal surface constituent relationship')} </pre>
SHAPE_ELEMENT_DERIVING_RELATIONSHIP	shape_aspect_deriving_relationship	47	302, 303	
shape_element_deriving_relationship to derived_shape_element (as relating_shape_element)	PATH		344, 333	<pre> shape_aspect_deriving_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => derived_shape_aspect </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
SHAPE_ELEMENT_LOCATING_RELATIONSHIP	dimensional_location	47	304, 305	
shape_element_locating_- relationship to shape_element (as located_shape_element)	PATH		335, 344	dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect
shape_element_locating_- relationship to shape_element (as locating_shape_element)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect
SHAPE_ELEMENT_RELATIONSHIP	shape_aspect_relationship	41	306	
description	descriptive_representation_- item.description	45		shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element relationship characterization'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'shape element relationship description'} representation.items[i] -> representation_item =>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				descriptive_representation_item descriptive_representation_item.description
name	descriptive_representation_- item.description	45		shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element relationship characterization'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'shape element relationship name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
shape_element_relationship to shape_element (as related_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect
shape_element_relationship to shape_element (as relating_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
SIMULTANEOUS_GEOMETRIC_TOLERANCE_REQUIREMENT	geometric_tolerance_group	210	340	geometric_tolerance_group <= property_definition {property_definition property_definition.description = 'simultaneous requirement'}
simultaneous_geometric_- tolerance_requirement to geometric_tolerance (as grouped_geometric_tolerances)	PATH		340	geometric_tolerance_group <= property_definition <- property_definition_relationship.relati- ng_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship property_definition_relationship.name = 'group geometric_tolerance'} property_definition_relationship.relati- ng_ property_definition -> property_definition => physical_unit_geometric_tolerance
SINGLE_BOUNDARY_DIMENSION	dimensional_size	45		
SINGLE_DATUM	physical_unit_datum	210	333	physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'single_datum'}
SINGULAR_DIMENSION_REPRESENTATION	shape_dimension_representation	47	312	
singular_dimension_- representation to data_element (as dimension_value)	PATH			shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit <- {measure_with_unit <- measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'}}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> measure_qualification.qualified[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {(type_qualifier.name = 'maximum dimension')} (type_qualifier.name = 'minimum dimension') (type_qualifier.name = 'theoretically exact') </pre>
STATISTICAL_DIMENSIONAL_TOLERANCE	(statistical_dimensional_location) (statistical_dimensional_size)	210 210		<pre> (statistical_dimensional_location <= dimensional_location) (statistical_dimensional_size <= dimensional_size_property <= dimensional_size) [property_definition] </pre>
statistical_dimensional_tolerance to dimensional_characteristic (as alternate_dimension)	PATH		335, 344	<pre> (statistical_dimensional_location <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'dimensional location property'} property_definition <- property_definition_relationship.related_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'substitutable dimension'} property_definition_relationship.related_ property_definition -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition property_definition.description = 'dimensional location property'} property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location (statistical_dimensional_size <= dimensional_size_property <= property_definition <- property_definition_relationship.related_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'substitutable dimension'} property_definition_relationship.relatng_- property_definition -> property_definition => dimensional_size_property <= dimensional_size) (statistical_dimensional_location <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition </pre>
statistical_dimensional_tolerance to ee_requirement_occurrence (as statistical_control_requirement)	PATH		340	<pre> (statistical_dimensional_location <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition {property_definition property_definition.description = 'dimensional location property'} property_definition) (statistical_dimensional_size <= dimensional_size_property <= property_definition) <- property_definition.relationship.relatiing_- property_definition property_definition.relationship {property_definition.relationship property_definition.relationship.name = 'dimensional tolerance statistical control requirement'} property_definition.relationship.related_- property_definition -> property_definition => requirements_property </pre>
STATISTICAL_GEOMETRIC_- TOLERANCE	statistical_geometric_tolerance	210	352	<pre> statistical_geometric_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance </pre>
statistical_geometric_tolerance to geometric_tolerance (as alternate_tolerance)	PATH		352	<pre> statistical_geometric_tolerance <= physical_unit_geometric_tolerance <= geometric_tolerance <- geometric_tolerance.relationship.related_- geometric_tolerance geometric_tolerance.relationship {geometric_tolerance.relationship geometric_tolerance.relationship.name = 'substitutable geometric_tolerance'} geometric_tolerance.relationship.relatiing_- geometric_tolerance -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
statistical_geometric_tolerance to ee_requirement_occurrence (as statistical_control_requirement)	PATH		340	<pre> geometric_tolerance => physical_unit_geometric_tolerance statistical_geometric_tolerance <= physical_unit_geometric_tolerance <= property_definition <- property_definition_relationship.relationship.relatin- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'geometric_tolerance statistical control requirement'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property </pre>
STRAIGHTNESS_TOLERANCE	physical_unit_geometric_tolerance	210	352	<pre> physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'straightness'} </pre>
STRATUM_FEATURE	stratum_feature	210	333, 322	<pre> stratum_feature <= shape_aspect </pre>
feature_of_size	descriptive_representation_item	45	342	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation.items[i] -> </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item representation_item.name = 'feature of size' representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}}
name	shape_aspect.name	41		stratum_feature <= shape_aspect shape_aspect.name
stratum_feature to stratum (as resident_stratum)	PATH		340, 265, 266, 338	stratum_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => stratum
SYMMETRY_TOLERANCE	symmetry_tolerance	210	352	symmetry_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'symmetry'}
TANGENT_PLANE	tangent	47	333	{tangent <= derived_shape_aspect <=

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect shape_aspect.description = 'plane'
TANGENT_SHAPE_ELEMENT	tangent	47		
THEORETICALLY_EXACT_DIMENSION_REPRESENTATION	shape_dimension_representation	47	342	{shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit <= measure_qualification.qualified_measure measure_qualification {measure_qualification.name = 'dimension value qualifier'} measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier {type_qualifier.name = 'theoretically exact'}}
TOLERANCE_SPECIFIC_RE- STRAINT_CONDITION	property_definition	41	361	{property_definition property_definition.description = 'tolerance specific restraint'}
tolerance_specific_restraint_ condition to dimensional_characteristic (as affected_tolerances)	PATH		344	property_definition <= property_definition_relationship.relati- ng_ property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'affected tolerance'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_relationship.related_ property_definition -> {property_definition property_definition.description = 'dimensional location property'} property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => dimensional_location) (property_definition => dimensional_size_property <= dimensional_size) </pre>
tolerance_specific_restraint_ - condition to geometric_tolerance (as affected_tolerances)	PATH			<pre> property_definition <- property_definition_relationship.related_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'affected_tolerance'} property_definition_relationship.related_ - property_definition -> property_definition => physical_unit_geometric_tolerance </pre>
TOLERANCE_ZONE	tolerance_zone	47	366	
tolerance_zone to tolerance_zone_boundary (as zone_boundary)	PATH		333	<pre> tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition tolerance_zone_definition.boundaries[i] -> shape_aspect => tolerance_zone_boundary </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
TOLERANCE_ZONE_AND_VIEWING_PLANE_ORIENTING_RELATIONSHIP	shape_aspect_relationship	41	362	{shape_aspect_relationship shape_aspect_relationship.name = 'tolerance zone and viewing plane'}
tolerance_zone_and_viewing_plane_orienting_relationship to viewing_plane (as orienting_viewing_plane)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => viewing_plane
tolerance_zone_and_viewing_plane_orienting_relationship to tolerance_zone (as oriented_tolerance_zone)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => tolerance_zone
TOLERANCE_ZONE_BOUNDARY	tolerance_zone_boundary	210	333	tolerance_zone_boundary <= shape_aspect {[shape_aspect <- tolerance_zone_definition.boundaries[i] tolerance_zone_definition] [shape_aspect.product_definitional = .FALSE.]}
TOLERANCE_ZONE_BOUNDARY_DEFINITION	property_definition	41	363	{property_definition (property_definition.description = 'boundary zone definition with specified size') (property_definition.description = 'conical tolerance zone boundary')}
tolerance_zone_boundary_definition to tolerance_zone_boundary (as defined_zone_boundary)	PATH		22, 333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => tolerance_zone_boundary
TOLERANCE_ZONE_BOUNDARY_MEMBER	shape_aspect	41	364	{shape_aspect [shape_aspect.description = 'tolerance zone boundary member'] [shape_aspect.product_definitional = .FALSE.]}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
TOLERANCE_ZONE_BOUNDARY_MEMBER_OFFSET	property_definition	41	365	{property_definition property_definition.description = 'tolerance zone boundary member offset'}
type_of_offset	property_definition.name	41		property_definition property_definition.name {(property_definition.name = 'towards material') (property_definition.name = 'away from material')}
tolerance_zone_boundary_- member_offset to tolerance_zone_boundary_- member (as offset_tolerance_zone_- boundary_member)	PATH			property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect [shape_aspect.description = 'tolerance zone boundary member'] [shape_aspect.product_definitional = .FALSE.]}
tolerance_zone_boundary_- member_offset to length_data_element (as offset_value)	PATH		109	property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'tolerance zone boundary member offset value'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
TOLERANCE_ZONE_CIRCULAR_OR_CYLINDRICAL_- OR_SPHERICAL_BOUNDARY	tolerance_zone_boundary	210	333	tolerance_zone_boundary <= shape_aspect {shape_aspect shape_aspect.description = 'circular or cylindrical or spherical'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
TOLERANCE_ZONE_DEFINITION	tolerance_zone_definition	47	367	
tolerance_zone_definition to tolerance_zone (as defined_tolerance_zone)	PATH			tolerance_zone_definition tolerance_zone_definition.zone -> tolerance_zone
tolerance_zone_definition to tolerance_zone_boundary_- definition (as boundary_definition)	PATH		333	tolerance_zone_definition tolerance_zone_definition.boundaries[i] -> {shape_aspect => tolerance_zone_boundary} shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition (property_definition.description = 'boundary zone definition with specified size') (property_definition.description = 'conical tolerance zone boundary')}
tolerance_zone_definition to tolerance_zone_form (as form)	PATH			tolerance_zone_definition tolerance_zone_definition.zone -> tolerance_zone tolerance_zone.form -> tolerance_zone_form
TOLERANCE_ZONE_DEFINITION_WITH_PER_UNIT_SIZE_SPECIFICATION	tolerance_zone_definition	47	368	
tolerance_zone_definition_with_- per_unit_size_specification to tolerance_zone_per_unit_size_- specification (as per_unit_size_specification)	PATH		333	tolerance_zone_definition tolerance_zone_definition.zone -> tolerance_zone <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition property_definition.description = 'tolerance zone per unit size'}
TOLERANCE_ZONE_DEFINITION_WITH_SPECIFIED_ORIENTATION	property_definition	41	369, 333	{property_definition [property_definition.description = 'tolerance zone orientation'] [property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => tolerance_zone]}
tolerance_zone_definition_with_- specified_orientation to datum_axis_related_orientation (as zone_orientation)	PATH			property_definition <- property_definition.relationship.relatating_- property_definition property_definition.relationship {property_definition.relationship property_definition.relationship.name = 'zone orientation'} property_definition.relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'datum axis related orientation'}
tolerance_zone_definition_with_- specified_orientation to viewing_plane_based_orientation (as zone_orientation)	PATH			property_definition <- property_definition.relationship.relatating_- property_definition property_definition.relationship {property_definition.relationship property_definition.relationship.name = 'zone orientation'} property_definition.relationship.related_- property_definition -> property_definition

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition property_definition.description = 'viewing plane based orientation'}
tolerance_zone_definition_with_- specified_orientation_to datum_based_vector_orientation (as zone_orientation)	PATH			property_definition <- property_definition_relationship.relatin_g_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'zone orientation'} property_definition_relationship.related_- property_definition -> property_definition {property_definition property_definition.description = 'datum based vector orientation'}
TOLERANCE_ZONE_EX- PLICIT_OPPOSING_BOUND- ARY_SET	tolerance_zone_explicit_- opposing_boundary_set	210	333	tolerance_zone_explicit_opposing_boundary_set <= tolerance_zone_boundary <= shape_aspect tolerance_zone_explicit_opposing_boundary_set <= tolerance_zone_boundary <= shape_aspect <- shape_aspect_relationship.relatin_g_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'zone boundary member'}
tolerance_zone_explicit_- opposing_boundary_set to tolerance_zone_boundary_- member (as boundary_members)	PATH		333	

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect [shape_aspect.description = 'tolerance zone boundary member'] [shape_aspect.product_definitional = .FALSE.]}
TOLERANCE_ZONE_FORM	tolerance_zone_form	47	370	
name	tolerance_zone_form.name	47		
tolerance_zone_form to tolerance_zone (as characterized_tolerance_zone)	PATH			tolerance_zone_form <- tolerance_zone.form tolerance_zone
TOLERANCE_ZONE_IM- PLICIT_OPPOSING_BOUND- ARY_SET	tolerance_zone_implicit_ opposing_boundary_set	210	333	tolerance_zone_implicit_opposing_boundary_set <= tolerance_zone_boundary <= shape_aspect
TOLERANCE_ZONE_OPPOS- ING_BOUNDARY_SET	(tolerance_zone_explicit_ opposing_boundary_set) (tolerance_zone_implicit_ opposing_boundary_set)	210 210		(tolerance_zone_explicit_opposing_boundary_set <= tolerance_zone_boundary <= shape_aspect) (tolerance_zone_implicit_opposing_boundary_set <= tolerance_zone_boundary <= shape_aspect)
TOLERANCE_ZONE_PER_ UNIT_SIZE_SPECIFICATION	property_definition	41	371	{property_definition property_definition.description = 'tolerance zone per unit size'}
tolerance_zone_per_unit_size_ specification to tolerance_zone (as specified_tolerance_zone)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => tolerance_zone
tolerance_zone_per_unit_size_- specification to length_data_element (as per_unit_size)	PATH		109	property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'per unit size'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
TOTAL_RUNOUT_TOLER- ANCE	total_runout_tolerance	210	352	total_runout_tolerance <= geometric_tolerance_with_specified_datum_- system <= physical_unit_geometric_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'total runout'}
TRUE_PROFILE	property_definition	41	377	{property_definition property_definition.description = 'true profile'}
true_profile to physical_feature_or_part_template (as profiled_surface)	PATH		333	property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect {(shape_aspect shape_aspect.product_definitional = .TRUE.) (shape_aspect => part_template_definition)} </pre>
true_profile to dimension_set (as profile_representation)	PATH			<pre> property_definition <- property_definition_relationship.relatin_ - property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'true profile specification'} property_definition_relationship.related_ - property_definition -> property_definition {property_definition property_definition.description = 'dimension set'} </pre>
true_profile to ee_document (as profile_representation)	PATH			<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> (document) (document => ee_specification) </pre>
true_profile to ee_rule_function (as profile_representation)	PATH		342	<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation => rule_function_definition </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
true_profile to bound_volume_shape (as profile_representation)	PATH		111	<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = '3d bound volume shape'} representation => shape_representation </pre>
USER_DEFINED_DIMEN- SION_QUALIFIER	measure_qualification	45	342, 378	<pre> {[measure_qualification measure_qualification.name = 'user defined dimension qualifier'] [measure_qualification measure_qualification.qualified_measure -> measure_with_unit => measure_representation_item <= representation_item <- representation.items[i] representation => shape_representation => shape_dimension_representation <- dimensional_characteristic_ representation.representation dimensional_characteristic_representation dimensional_characteristic_ representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_ location) (dimensional_characteristic = dimensional_ size)]}] </pre>
description	type_qualifier.name	45		<pre> measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				type_qualifier type_qualifier.name
name	measure_qualification.description	45		
USER_DEFINED_GEOMETRIC_TOLERANCE_QUALIFIER	measure_qualification	45	352, 379	{[measure_qualification measure_qualification.name = 'user defined geometric tolerance qualifier'] [measure_qualification measure_qualification.qualified_measure -> measure_with_unit <- geometric_tolerance.magnitude geometric_tolerance => physical_unit_geometric_tolerance]}
description	type_qualifier.name	45		measure_qualification measure_qualification.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name
name	measure_qualification.description	45		
VIEWING_PLANE	viewing_plane	210	333	viewing_plane <= shape_aspect {shape_aspect [shape_aspect.description = 'affected plane'] [shape_aspect.product_definitional = .FALSE.]}
VIEWING_PLANE_BASED_ORIENTATION	property_definition	41	386	{property_definition property_definition.description = 'viewing plane based orientation'}
viewing_plane_based_orientation to tolerance_zone_and_viewing_ plane_orienting_relationship (as orienting_relationship)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'tolerance zone and viewing plane'}

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
viewing_plane_based_orientation to viewing_plane_definition (as viewing_plane_specification)	PATH			<pre> property_definition <- property_definition.relationship.relatiing_- property_definition property_definition.relationship {property_definition.relationship.name = 'viewing_plane_specification'} property_definition.relationship.related_- property_definition -> property_definition {property_definition.description = 'viewing plane property'} </pre>
VIEWING_PLANE_DEFINI- TION	property_definition	41	387	<pre> {property_definition property_definition.description = 'viewing plane property'} </pre>
viewing_plane_definition to viewing_plane (as defined_viewing_plane)	PATH		333	<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => viewing_plane </pre>
viewing_plane_definition to axis_placement (as viewing_plane_representation)	PATH		109	<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement => axis1_placement </pre>

Table 50 – Mapping table for geometric_dimensioning_and_tolerance UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path

Table 51 – Mapping table for geometrically_bounded_2d_wireframe UoF

Application element	AIM element	Source	Rules	Reference path
GEOMETRICALLY_- BOUNDED_2D_WIREFRAME	geometrically_bounded_2d_- wireframe_representation	503		

Table 52 – Mapping table for inter_stratum_feature_template UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_TERMINATION_PASSAGE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= shape_aspect {shape_aspect shape_aspect.description = 'component termination passage template'}
component_termination_passage_- template to default_component_termination_- passage_definition (as of_passage_technology)	PATH		333	shape_aspect shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'default component termination passage definition'} shape_aspect => passage_technology
COMPONENT_TERMINATION_PASSAGE_TEMPLATE_- INTERFACE_TERMINAL	shape_aspect	41		{shape_aspect shape_aspect.description = 'component termination passage template interface terminal'}
COMPONENT_TERMINATION_PASSAGE_TEMPLATE_- JOIN_TERMINAL	shape_aspect	41		{shape_aspect shape_aspect.description = 'component termination passage template join terminal'}
component_termination_passage_- template_join_terminal to inter_stratum_extent (as disallowed_inter_stratum_extent)	PATH			shape_aspect <= shape_aspect_relationship.related_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'disallowed inter stratum extent'} shape_aspect_relationship.related_shape_- -> shape_aspect shape_aspect.of_shape -> product_definition_shape <=

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition { property_definition property_definition.description = 'finished stratum extent' } property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship { product_definition_relationship.name = 'inter stratum extent' } </pre>
COMPONENT_TERMINA- TION_PASSAGE_TEMPLATE_- TERMINAL	shape_aspect	41	54	<pre> [[shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal')]] [shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model' [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'[]]] </pre>
name	shape_aspect.name	41		
<pre> component_termination_passage_- template_terminal to component_termination_passage_- template (as associated_definition) </pre>	PATH		53, 333, 340	<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'access mechanism'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage template'} shape_aspect => part_template_definition </pre>
<pre> component_termination_passage_- template_terminal to connection_zone (as terminal_connection_zone) </pre>	PATH			<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>
COUNTERBORE_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect shape_aspect.description = 'counterbore passage template'} </pre>
counterbore_passage_template to unsupported_passage_template (as smaller_passage)	PATH		333, 340	<pre> part_template_definition <= shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect.relationship.description = 'smaller passage'} shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
counterbore_passage_template to blind_passage_template (as larger_passage)	PATH			<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect.relationship.description = 'larger passage'} </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
COUNTERSUNK_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect shape_aspect.description = 'countersunk passage template'} </pre>
countersunk_passage_template to unsupported_passage_template (as constant_diameter_passage)	PATH		333, 340	<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'constant diameter passage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
countersunk_passage_template to blind_passage_template (as tapered_passage)	PATH			<pre> part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'larger passage'} </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
INTER_STRATUM_FEATURE_TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'via template') (shape_aspect.description = 'component termination passage template') (shape_aspect.description = 'unsupported passage template') (shape_aspect.description = 'inter stratum feature edge template') (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'counterbore passage template') (shape_aspect.description = 'countersunk passage template') (shape_aspect.description = 'blind passage template')}} </pre>
inter_stratum_feature_template to passage_technology (as of_passage_technology)	PATH			<pre> shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship stratum_feature_passage_technology'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => passage_technology </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
UNSUPPORTED_PASSAGE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect (shape_aspect.description = 'unsupported passage template') (shape_aspect.description = 'counterbore passage template') (shape_aspect.description = 'countersunk passage template') (shape_aspect.description = 'blind passage template')}}}
unsupported_passage_template to default_unsupported_passage_ definition (as of_passage_technology)	PATH		333	shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.related_shape_ aspect -> {shape_aspect shape_aspect.description = 'default unsupported passage definition'} shape_aspect => passage_technology
VIA_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'via template'}
via_template to default_via_definition (as of_passage_technology)	PATH		333	shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.related_shape_ aspect ->

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect shape_aspect.description = 'default via definition'} shape_aspect => passage_technology</pre>
VIA_TEMPLATE_TERMINAL	shape_aspect	41	384, 385	<pre>{[shape_aspect shape_aspect.description = 'via template terminal'] [shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition [product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition']]}</pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
name	shape_aspect.name	41		
via_template_terminal to connection_zone (as terminal_connection_zone)	PATH			<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>
via_template_terminal to inter_stratum_extent (as disallowed_inter_stratum_extent)	PATH			<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'disallowed inter stratum extent'} shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition property_definition.description = 'finished stratum extent'} property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship </pre>

Table 52 – Mapping table for inter_stratum_feature_template UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'}</pre>
via_template_terminal to via_template (as associated_definition)	PATH		333, 340	<pre>shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'access mechanism'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'via template'} shape_aspect => part_template_definition</pre>

Table 53 – Mapping table for interconnect_bill_of_material UoF

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_PRODUCT	alternate_product_relationship	44		{alternate_product_relationship alternate_product_relationship.basis = 'alternate_product'}
alternate_product to ee_product (as defined_alternate_product)	PATH			alternate_product_relationship alternate_product_relationship.alternate -> product
ALTERNATE_SELECT_PROD- UCT	alternate_product_relationship	44		{alternate_product_relationship (alternate_product_relationship.basis = 'alternate_product') (alternate_product_relationship.basis = 'test select product') (alternate_product_relationship.basis = 'assembly alternate product')}
alternate_select_product to contract (as authorizing_agreement)	PATH			alternate_product_relationship contract_assigned_item = alternate_product_- relationship contract_assigned_item <= applied_contract_assignment.items[i] applied_contract_assignment <= contract_assignment contract_assignment.assigned_contract -> contract
alternate_select_product to ee_product (as defined_alternate_product)	PATH			alternate_product_relationship alternate_product_relationship.alternate -> product
alternate_select_product to ee_product (as defining_base_product)	PATH			alternate_product_relationship alternate_product_relationship.base -> product

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_ALTERNATE_- PRODUCT	alternate_product_relationship	44		{alternate_product_relationship alternate_product_relationship.basis = 'assembly alternate product'}
assembly_alternate_product to physical_unit_design_view (as basis_context)	PATH		14, 268, 338, 339, 354	alternate_product_relationship alternate_product_relationship.base -> product <- product_definition_formation.of_product product_definition_formation <- product_definition_formation product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
ASSEMBLY_COMPOSITION_- RELATIONSHIP	assembly_component_usage	44	354, 339, 15	{assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly composition'}
assembly_composition_- relationship to assembly_component (as component)	PATH		354, 339, 338, 340, 333	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition => component_definition} (product_definition [product_definition => component_definition] [product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect]) </pre>
assembly_composition_- relationship to ee_material (as assembly_bonding_agent)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- material_designation.definitions[i] material_designation </pre>
assembly_composition_- relationship to physical_unit_design_view (as assembly)	PATH		354, 339, 338	<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition </pre>

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design' product_definition => physical_unit </pre>
assembly_composition_- relationship to security_classification (as security_code)	PATH			<pre> assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
ASSEMBLY_MATERIAL_COM- POSITION_RELATIONSHIP	<pre> (next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage]) </pre>	44 44 44		<pre> (next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}) ([quantified_assembly_component_usage] [next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}]) </pre>

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
as_required	PATH	41		quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit measure_with_unit.value_component -> measure_value measure_value = descriptive_measure {(descriptive_measure = 'as required') (descriptive_measure = 'quantity specified')}]
assembly_material_composition_- relationship to data_element (as quantity)	PATH			quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit
assembly_material_composition_- relationship to material_definition (as component_material)	PATH			next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'material definition'}
assembly_material_composition_- relationship to physical_unit_design_view (as assembly)	PATH		338	next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition ->

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition product_definition => physical_unit
assembly_material_composition_- relationship to security_classification (as security_code)	PATH			next_assembly_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification
INTERCONNECT_MODULE_- STRATUM_ASSEMBLY_RELA- TIONSHIP	assembly_component_usage	44	192, 354, 339, 193	{assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'}
reference_designation	assembly_component_- usage.reference_designator	44		
interconnect_module_stratum_- assembly_relationship to interconnect_module (as assembly)	PATH		263, 264, 268, 267, 265, 266, 338	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference ->

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'physical design'}}
interconnect_module_stratum_- assembly_relationship to stratum (as component)	PATH			product_definition => physical_unit assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => stratum
MATERIAL_COMPOSITION_- RELATIONSHIP	product_material_composition_- relationship	45	207	
composition_basis	product_material_composition_- relationship.composition_basis	45		
composition_class	product_material_composition_- relationship.class	45		
determination_method	product_material_composition_- relationship.determination_method	45		
material_composition_relationship to data_element (as constituent_amount)	PATH			product_material_composition_relationship product_material_composition_- relationship.constituent_amount[i] -> measure_with_unit
material_composition_relationship to ee_material (as related_product)	PATH		339	product_material_composition_relationship <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition characterized_product_definition = product_- definition

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation
material_composition_relationship to ee_material (as relating_product)	PATH		339	product_material_composition_relationship <= product_definition_relationship product_definition_relationship.relatiing_ product_definition -> product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation
NEXT_HIGHER_ASSEMBLY_- RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44	213, 214	(next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=] assembly_component_usage)
reference_designation	PATH			next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator
next_higher_assembly_- relationship to data_element (as quantity)	PATH			next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage.quantity -> measure_with_unit

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect.description = 'interconnect module component stratum based terminal' (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PROMISSORY_USAGE_IN_PRODUCT_MODEL_RELATIONSHIP	applied_promissory_usage_in_product_model_assignment	210		{applied_promissory_usage_in_product_model_assignment <= group_assignment group_assignment.assigned_group -> group => promissory_usage_in_product_model}
promissory_usage_in_product_model_relationship to Ee_product_definition (as component)	PATH			applied_promissory_usage_in_product_model_assignment applied_promissory_usage_in_product_model_assignment.items[i] -> promissory_usage_in_product_model_assigned_item promissory_usage_in_product_model_assigned_item = product_definition (product_definition => product_definition_with_associated_documents) (product_definition) (product_definition => evaluation_product_definition) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition => analytical_model_definition) (product_definition =>

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				footprint_definition) (product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => stratum) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit)
promissory_usage_in_product_- model_relationship to ee_product_model (as product_model)	PATH			applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_concept
promissory_usage_in_product_- model_relationship to security_classification (as security_code)	PATH			applied_promissory_usage_in_product_model_- assignment security_classification_assigned_item = applied_promissory_usage_in_product_model_- assignment security_classification_assigned_item <= applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment

Table 53 – Mapping table for interconnect_bill_of_material UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment.assigned_ security_classification -> security_classification
PROMISSORY_USAGE_RELATIONSHIP promissory_usage_relationship to physical_unit_design_view (as assembly)	promissory_usage_occurrence PATH	44	275, 276	promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
promissory_usage_relationship to security_classification (as security_code)	PATH			promissory_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <= applied_security_classification_ assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification
SUBSTITUTE_PRODUCT	product	41	331	
substitute_product to assembly_ composition_relationship (as substitutable_component)	PATH		268, 339, 354	product <= product_definition_formation.of_product product_definition_formation <= product_definition_formation product_definition <=

Table 53 – Mapping table for interconnect_bill_of_material UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'assembly_composition'} product_definition_relationship => product_definition_usage => assembly_component_usage {assembly_component_usage <- assembly_component_usage_substitute.base} </pre>
substitute_product to ee_product (as defined_alternate_product)	PATH			<pre> product <- product_definition_formation.of_product product_definition_formation <- product_definition_formation product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship => product_definition_usage => assembly_component_usage <- substitute_substitute} product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition_formation -> product_definition_formation product_definition_formation.of_product -> product </pre>
TEST_SELECT_PRODUCT	alternate_product_relationship	44		<pre> {alternate_product_relationship alternate_product_relationship.basis = 'test select product'} </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF

Application element	AIM element	Source	Rules	Reference path
ARRAY_PLACEMENT_GROUP	(array_placement_group - component_definition) (array_placement_group - component_shape_aspect)	210 210	338, 333	(array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <= product_definition_relationship.relatiing_ product_definition {product_definition_relationship product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_ product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']}) (array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.relatiing_shape_ aspect {shape_aspect_relationship shape_aspect_relationship.name = 'group component'})

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect
ASSEMBLY_COMPONENT_- PLACEMENT_LINK	(assembly_group_component_- shape_aspect_placement_link) (assembly_group_component_- definition_placement_link)	210 210		({[assembly_group_component_shape_aspect_- placement_link <= shape_aspect] [assembly_group_component_shape_aspect_- placement_link <= shape_aspect_relationship]}) ({[assembly_group_component_definition_- placement_link <= product_definition] [assembly_group_component_definition_- placement_link <= product_definition_relationship]})
assembly_component_- placement_link to assembly_component (as precedent_element)	PATH			(assembly_group_component_shape_aspect_- placement_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect => component_shape_aspect) (assembly_group_component_definition_- placement_link <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition)
assembly_component_- placement_link to assembly_component (as subsequent_element)	PATH			(assembly_group_component_shape_aspect_- placement_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect)

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(assembly_group_component_definition_ - placement_link <= product_definition_relationship product_definition_relationship.related_ - product_definition -> product_definition => component_definition)</pre>
ASSEMBLY_GROUP_COMPONENT	<pre>(assembly_group_component_ - definition) (assembly_group_component_ - shape_aspect)</pre>	210 210	338, 333	<pre>(assembly_group_component_definition <= component_definition <= product_definition {product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_ - product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']}) (assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect {[shape_aspect <- shape_aspect_relationship.related_shape_ - aspect {shape_aspect_relationship shape_aspect_relationship.name = 'group component' }</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect] [shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => component_definition => assembly_group_component_definition]] </pre>
INTERFACED_GROUP_COMPONENT	interfaced_group_component_definition	210		<pre> interfaced_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition </pre>
interfaced_group_component to interface_component (as group_interface_component)	PATH		338, 333	<pre> interfaced_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <- product_definition_relationship.relatiing_- product_definition {product_definition_relationship product_definition_relationship.name = 'group interface component'} product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'physical occurrence' product_definition => component_definition => interface_component_definition </pre>
LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT	<pre> (linear_array_placement_group_ - component_definition) (linear_array_placement_group_ - component_shape_aspect) </pre>	210 210		<pre> (linear_array_placement_group_component_ - definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition {product_definition <- product_definition_relationship.relationship_ - product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_ - product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'[]}) (linear_array_placement_group_component_ - shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => component_definition} {shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect => component_shape_aspect} {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence'] [product_definition product_definition.description = 'laminare component']]})
linear_array_placement_group_ - component to assembly_component_ - placement_link (as element)	PATH			(linear_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <= product_definition.relationship.relatiing_ product_definition {product_definition.relationship product_definition.relationship.name = 'element'} product_definition.relationship.related_ product_definition -> product_definition => assembly_group_component_definition_ placement_link) (linear_array_placement_group_component_ shape_aspect <= array_placement_group_component_shape_aspect <=

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_group_component_shape_aspect_- placement_link) (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (linear_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) </pre>
linear_array_placement_group_- component to length_data_element (as initial_node_positive_x_- displacement)	PATH			<pre> assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_group_component_shape_aspect_- placement_link) (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (linear_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>property_definition.definition property_definition <- {property_definition placement_group_properties'} property_definition.definition property_definition_representation.used_ - representation -> representation {[representation representation.name = 'linear array properties'] [representation representation.context_of_items -> representation_context [representation_context => global_unit_assigned_context] [representation_context => global_uncertainty_assigned_context] [representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']} representation.items[i] -> representation_item => {representation_item</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item.name = 'initial element positive x displacement' measure_representation_item <= {measure_with_unit length_measure_with_unit} measure_with_unit.value_component </pre>
linear_array_placement_group_- component to length_data_element (as location_uncertainty)	PATH			<pre> (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (linear_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) property_definition.definition property_definition <- {property_definition property_definition.name = 'linear array placement group properties'} </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation.used_ representation -> representation {representation representation.name = 'linear array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_- context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit </pre>
linear_array_placement_group_- component to length_data_element (as terminus_node_positive_x_- displacement)	PATH			<pre> (linear_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>characterized_definition <-> (linear_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-> property_definition.definition property_definition <- {property_definition property_definition.name = 'linear array placement group properties'} property_definition_representation.definition property_definition_representation.used_- representation -> representation {[representation representation.name = 'linear array properties'] [representation representation.context_of_items -> representation_context [representation_context representation_context => global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_context => parametric_representation_context[] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context'] representation.items[i] -> representation_item => {representation_item representation_item.name = 'terminus element positive x displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component </pre>
LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_LINK	(linear_array_component_shape_- aspect_link) (linear_array_component_- definition_link)	210 210		<pre> ({[linear_array_component_shape_aspect_link <= shape_aspect] [linear_array_component_shape_aspect_link <= shape_aspect_relationship]}) ({[linear_array_component_definition_link <= product_definition] [linear_array_component_definition_link <= product_definition_relationship]}) </pre>
linear_array_placement_group_- component_link to linear_array_placement_group_- component (as precedent_element)	PATH			<pre> (linear_array_component_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect => linear_array_placement_group_component_shape_- aspect) </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(linear_array_component_definition_link <= product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => component_definition => assembly_group_component_definition => array_placement_group_component_definition => linear_array_placement_group_component_ definition)</pre>
linear_array_placement_group_- component_link to linear_array_placement_group_- component (as subsequent_element)	PATH			<pre>(linear_array_component_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect => linear_array_placement_group_component_shape_- aspect) (linear_array_component_definition_link <= product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => component_definition => assembly_group_component_definition => array_placement_group_component_definition => linear_array_placement_group_component_ definition)</pre>
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		<pre>(component_terminal <= shape_aspect {shape_aspect</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)</pre>
RECTANGULAR_ARRAY_- PLACEMENT_GROUP_COM- PONENT	(rectangular_array_placement_- group_component_definition) (rectangular_array_placement_- group_component_shape_aspect)	210 210		<pre>(rectangular_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <- product_definition_relationship.relationship_- product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_- product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']}) (rectangular_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <=</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>shape_aspect { shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => component_definition { shape_aspect <- shape_aspect_relationship.relati shape_aspect { shape_aspect_relationship shape_aspect_relationship.name = 'group component' shape_aspect_relationship shape_aspect_relationship.related_shape_ -> shape_aspect { shape_aspect => component_shape_aspect } shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition } } }</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence'] [product_definition product_definition.description = 'laminare component']]})
rectangular_array_placement_ group_component to linear_array_placement_group_ component_link (as element)	PATH			(rectangular_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition <= product_definition.relationship.relatiing_ product_definition {product_definition.relationship product_definition.relationship.name = 'element'} product_definition.relationship.related_ product_definition -> product_definition => linear_array_component_definition_link) (rectangular_array_placement_group_component_ shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <=

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => linear_array_component_shape_aspect_link) (rectangular_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (rectangular_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) property_definition.definition property_definition <- </pre>
rectangular_array_placement_- group_component to length_data_element (as initial_row_positive_y_- displacement)	PATH			<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => linear_array_component_shape_aspect_link) (rectangular_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (rectangular_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) property_definition.definition property_definition <- </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{property_definition property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context => global_unit_assigned_context] [representation_context => global_uncertainty_assigned_context] [representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']} representation_items[i] -> representation_item {representation_item.name = 'initial row positive y displacement'}</pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value component (rectangular_array_placement_group_component_ definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <-) (rectangular_array_placement_group_component_ shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) property_definition.definition property_definition <- {property_definition {property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> </pre>
rectangular_array_placement_ group_component to length_data_element (as location_uncertainty)	PATH			

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation representation.name = 'rectangular array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_- context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit } (rectangular_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (rectangular_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition </pre>
rectangular_array_placement_- group_component to length_data_element (as terminus_row_positive_y_- displacement)	PATH			<pre> representation { representation representation.name = 'rectangular array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_- context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit } (rectangular_array_placement_group_component_- definition <= array_placement_group_component_definition <= assembly_group_component_definition <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <-) (rectangular_array_placement_group_component_- shape_aspect <= array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>characterized_definition <- property_definition.definition property_definition <- {property_definition property_definition.name = 'rectangular array placement group properties'} property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context => global_unit_assigned_context] [representation_context global_uncertainty_assigned_context] [representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']}] representation_items[i] -> representation_item =></pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'terminus row positive y displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component </pre>
SUB_ASSEMBLY_RELATIONSHIP	(assembly_component_usage) (shape_aspect_relationship)	44 41		
sub_assembly_reference_designation	(assembly_component_usage.reference_designator) (shape_aspect_relationship.description)	44 41		
sub_assembly_relationship to assembly_component (as related_component)	PATH			<pre> (assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'}) product_definition_relationship.related_ product_definition -> product_definition => component_definition (shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'group component'}) shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect) </pre>

Table 54 – Mapping table for interconnect_component_grouping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
sub_assembly_relationship to assembly_group_component (as relating_component)	PATH		338	(assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship.relati- ng_<- product_definition -> product_definition => component_definition => assembly_group_component_definition) (shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship.relati- ng_shape_- aspect -> shape_aspect => component_shape_aspect => assembly_group_component_shape_aspect)

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_FEATURE_TO_- PHYSICAL_USAGE_VIEW_AS- SIGNMENT	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'component feature to physical usage assignment'}
component_feature_to_physical_- usage_view_assignment to component_feature (as assigned_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => (component_terminal) (laminated_component_interface_terminal) (component_feature)
component_feature_to_physical_- usage_view_assignment to part_feature (as defined_part_feature)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface')

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
COMPONENT_TERMINAL_- TO_INTERCONNECT_MOD- ULE_INTERFACE_TERMI- NAL_ASSIGNMENT	shape_aspect_relationship	41		<pre> {shape_aspect_relationship [shape_aspect_relationship.name = 'component feature to physical usage assignment'] [shape_aspect_relationship.description = 'component terminal to interconnect module interface terminal assignment']} </pre>

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- interconnect_module_terminal_- assignment to component_termination_passage_- interface_terminal (as assigned_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage interface terminal'} shape_aspect => component_shape_aspect
component_terminal_to_- interconnect_module_terminal_- assignment to interconnect_module_interface_- terminal (as usage_view_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'interconnect module'} shape_aspect => interconnect_module_terminal => interconnect_module_interface_terminal
component_terminal_to_- interconnect_module_interface_- terminal_assignment to component_termination_passage_- interface_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage interface terminal'} shape_aspect => laminante_component_interface_terminal

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- interconnect_module_interface_- terminal_assignment to land_interface_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'land interface terminal'} shape_aspect => laminar_component_interface_terminal
component_terminal_to_- interconnect_module_interface_- terminal_assignment to non_functional_land_interface_- terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'non functional land interface terminal'} shape_aspect => laminar_component_interface_terminal
component_terminal_to_- interconnect_module_interface_- terminal_assignment to printed_connector_component_- interface_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'printed connector component interface terminal'} shape_aspect => laminar_component_interface_terminal
COMPONENT_TERMINAL_- TO_INTERCONNECT_MOD- ULE_JOIN_TERMINAL_AS- SIGNMENT	shape_aspect_relationship	41		{shape_aspect_relationship [shape_aspect_relationship.name = 'component feature to physical usage assignment'] [shape_aspect_relationship.description = 'component terminal to interconnect module join terminal assignment']}]
component_terminal_to_- interconnect_module_join_- terminal_assignment to component_termination_passage_- join_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_shape_aspect

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- interconnect_module_join_- terminal_assignment to interconnect_module_join_- terminal (as usage_view_terminal)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition product_definition.name = 'interconnect module'} shape_aspect => interconnect_module_terminal => interconnect_module_join_terminal </pre>
component_terminal_to_- interconnect_module_join_- terminal_assignment to interconnect_module_macro_- component_join_terminal (as assigned_terminal)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect module component terminal'} shape_aspect => component_terminal => interconnect_module_macro_component_join_- terminal </pre>

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_terminal_to_- interconnect_module_join_- terminal_assignment to land_join_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'land join terminal'} shape_aspect => component_shape_aspect
component_terminal_to_- interconnect_module_join_- terminal_assignment to printed_component_join_terminal (as assigned_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'printed component join terminal'} shape_aspect => component_shape_aspect
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal)	210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') })
STRATUM_CONCEPT_TO_- PHYSICAL_USAGE_VIEW_AS- SIGNMENT	shape_aspect_relationship	41	316	{shape_aspect_relationship shape_aspect_relationship.name = 'stratum concept to physical usage view assignment'}
stratum_concept_to_physical_- usage_view_assignment to part_feature (as defined_part_feature)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'])

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect =>

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				packaged_part_terminal) (shape_aspect => package_body_surface}}
stratum_concept_to_physical_- usage_view_assignment to inter_stratum_feature (as assigned_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature
stratum_concept_to_physical_- usage_view_assignment to laminare_component (as assigned_design_object)	PATH		333, 340, 268, 338	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminare text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect {(shape_aspect.description = 'laminare component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions')

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component') (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition {product_definition.description = 'laminated component'}) shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect !{(shape_aspect => stratum_surface) (shape_aspect => stratum_feature)}} </pre>
stratum_concept_to_physical_ - usage_view_assignment to stratum (as assigned_design_object)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect !{(shape_aspect => stratum_surface) (shape_aspect => stratum_feature)}} </pre>

Table 55 – Mapping table for interconnect_design_information_assignment_to_usage_view UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => stratum
stratum_concept_to_physical_- usage_view_assignment to stratum_feature (as assigned_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature
stratum_concept_to_physical_- usage_view_assignment to stratum_surface (as assigned_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_surface

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_MODULE	physical_unit	210	268	<pre> physical_unit <= product_definition {product_definition.name = 'assembly module'} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] </pre>
CONNECTIVITY_ALLOCATION_TO_PHYSICAL_NETWORK	property_definition_relationship	41		<pre> {property_definition_relationship property_definition_relationship.name = 'connectivity allocation to physical network'} </pre>
connectivity_allocation to aggregate_connectivity_- requirement (as connectivity_requirement)	PATH			<pre> property_definition_relationship property_definition_relationship.relatin_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship {product_definition_relationship.name = 'aggregate connectivity requirement'} </pre>
connectivity_allocation to physical_network (as implementation)	PATH			<pre> property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition => </pre>

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape <- shape_aspect.of_shape shape_aspect => physical_network
INTERCONNECT_MODULE	physical_unit	210	338, 268	physical_unit <= product_definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}}}
interconnect_module to assembly_module (as assembly_design_requirement)	PATH			physical_unit <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'assembly_design_requirement'} product_definition_relationship.related_ product_definition -> product_definition {product_definition [product_definition => physical_unit] [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}}}

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PART_CONNECTED_TERMINALS_DEFINITION	part_connected_terminals_definition	210		part_connected_terminals_definition <= shape_aspect
name	shape_aspect.name	41	333	part_connected_terminals_definition <= shape_aspect shape_aspect.name
part_connected_terminals_definition to part_terminal (as connected_terminals)	PATH		333	part_connected_terminals_definition <= shape_aspect <- shape_aspect.relationship.relatng_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'member connected_terminal'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal) (interconnect_module_terminal) (packaged_part_terminal)
part_connected_terminals_definition to Interconnect_module_usage_view (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition definition {product_definition [product_definition.name = 'interconnect module']}

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PHYSICAL_NODE_REQUIREMENT_TO_IMPLEMENTING_COMPONENT_ALLOCATION	physical_node_requirement_to_implementing_component_allocation	210		{[physical_node_requirement_to_implementing_component_allocation <= shape_aspect] [physical_node_requirement_to_implementing_component_allocation <= shape_aspect relationship]}
physical_node_requirement_to_implementing_component_allocation to assembly_component (as implementation_component)	PATH		338, 340, 333	physical_node_requirement_to_implementing_component_allocation <= shape_aspect.relationship.related_shape_aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition {[product_definition => component_definition] [product_definition.description != 'laminated component']}]
physical_node_requirement_to_implementation_component_allocation to assembly_joint (as implementation_joint_set)	PATH			physical_node_requirement_to_implementation_component_allocation <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'implementation_joint'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint
physical_node_requirement_to_implementation_component_allocation to interface_mounted_join (as implementation_interface_joint_set)	PATH			physical_node_requirement_to_implementation_component_allocation <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'implementation_interface_joint'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => interface_mounted_join

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_node_requirement_to_implementation_component_allocation to physical_connectivity_definition (as physical_node_requirement)	PATH			physical_node_requirement_to_implementation_component_allocation <= shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => physical_connectivity_definition
TEST_REQUIREMENT_ALLOCATION	requirement_allocation_group	210		{[requirement_allocation_group <= property_definition_relationship][property_definition_relationship property_definition_relationship.name = 'test requirement_allocation'] [requirement_allocation_group <= group]} requirement_allocation_group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = product_definition_formation}
test_requirement_allocation to ee_product_version (as target_product_object)	PATH			requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product_definition_formation
test_requirement_allocation to test_requirement_occurrence (as allocated_requirement)	PATH			requirement_allocation_group <= property_definition_relationship property_definition_relationship.relatng_shape_aspect -> property_definition => {property_definition.description = 'test requirements_property'

Table 56 – Mapping table for interconnect_functional_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
test_requirement_allocation to part_feature_model_port_- assignment (as analytical_model_based_test_- access)	PATH			<pre> requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> property_definition_representation {property_definition_representation property_definition_representation.name = 'part feature model port assignment'}</pre>
test_requirement_allocation to functional_usage_view_to_part_- terminal_assignment (as functional_specification_- based_test_access)	PATH			<pre> requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'functional terminal allocation'}</pre>

Table 57 – Mapping table for interconnect_interface_component UoF

Application element	AIM element	Source	Rules	Reference path
EE_SPECIFICATION	ee_specification	210		ee_specification <= document
ee_specification to organization (as source)	PATH			<pre> ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_role organization_role.name = 'document source'} organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} </pre>

Table 57 – Mapping table for interconnect_interface_component UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PRINTED_CONNECTOR_COM- PONENT	printed_component	210	338, 268	printed_component <= component_definition <= product_definition {product_definition [product_definition.name = 'printed connector component'] [product_definition.description = 'printed component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']}]
printed_connector_component to printed_connector_template (as usage_definition)	PATH		338, 340, 333	printed_component <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition

Table 57 – Mapping table for interconnect_interface_component UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'printed connector template'} shape_aspect => part_template_definition</pre>

Table 58 – Mapping table for interconnect_marking UoF

Application element	AIM element	Source	Rules	Reference path
CONFIGURATION_CONTROLLED_MESSAGE	descriptive_representation_item	45	60	
message_content	descriptive_representation_item.description	45		
message_type_specification	representation_item.name	43	109	descriptive_representation_item <= representation_item representation_item.name {(representation_item.name = 'approval date') (representation_item.name = 'manufacturer code') (representation_item.name = 'part number') (representation_item.name = 'revision identifier')}}
configuration_controlled_message to ee_approval (as message_source)	PATH		13	descriptive_representation_item <= representation_item <- representation.items[i] representation approval_assigned_item = representation approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
configuration_controlled_message to ee_product_version (as message_source)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_representation property_definition_representation

Table 58 – Mapping table for interconnect_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.formformation -> product_definition_formation
configuration_controlled_message to organization (as message_source)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] representation organization_assigned_item = representation organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization
EE_SPECIFICATION	ee_specification	210		ee_specification <= document
ee_specification to organization (as source)	PATH			ee_specification <= document organization_assigned_item = document organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment

Table 58 – Mapping table for interconnect_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> organization_assignment.role -> organization_role organization_role.name = 'document source' organization_assignment.assigned_organization -> organization </pre>
ee_specification to person_and_organization (as source)	PATH			<pre> ee_specification <= document person_and_organization_assigned_item = document person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'document source'} person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
MARKING	representation	43	206	<pre> {representation representation.name = 'marking'} representation <- [property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 58 – Mapping table for interconnect_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect[]
marking to axis_placement (as placement)	PATH			representation representation.items[i] -> {representation_item representation_item.name = 'marking location'} representation_item => geometric_representation_item => placement
marking to ee_text (as marked_message)	PATH			representation representation.items[i] -> representation_item => descriptive_representation_item
marking to ee_material (as marking_material)	PATH			representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> {characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect} characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_ designation

Table 58 – Mapping table for interconnect_marking UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'marking material'} </pre>
marking to ee_specification (as marking_specification)	PATH			<pre> representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document <- document_usage_constraint.source document_usage_constraint document_usage_constraint.subject_element = 'marking specification'} document => ee_specification </pre>
marking to physical_unit_3d_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> representation {representation.name = '3d bound volume shape'} </pre>
marking to physical_unit_planar_shape (as marked_device_shape)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'marked device shape'} representation_relationship.rep_2 -> </pre>

Table 58 – Mapping table for interconnect_marking UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation {representation.name = 'planar projected shape'}
marking to feature_shape_definition (as message_representation)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'}
marking to styled_text (as message_representation_text)	PATH			representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'message representation'} representation_relationship.rep_2 -> representation {representation.description = 'text representation'} representation_items[i] -> representation_item representation_item.name = 'marking text' representation_item => (geometric_representation_item => text_literal) (styled_item => annotation_occurrence => annotation_text_occurrence)

Table 59 – Mapping table for interconnect_module_connection_routing UoF

Application element	AIM element	Source	Rules	Reference path
CONDUCTIVE_INTERCONNECT_ELEMENT_TERMINAL_LINK	conductive_interconnect_element_terminal_link	210		{[conductive_interconnect_element_terminal_link <= shape_aspect] [conductive_interconnect_element_terminal_link <= shape_aspect_relationship]}
conductive_interconnect_element_terminal_link to conductive_interconnect_element_terminal (as precedent_point)	PATH			conductive_interconnect_element_terminal_link <= shape_aspect_relationship shape_aspect_relationship.relying_shape_aspect -> shape_aspect => component_terminal
conductive_interconnect_element_terminal_link to conductive_interconnect_element_terminal (as subsequent_point)	PATH			conductive_interconnect_element_terminal_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_terminal
conductive_interconnect_element_terminal_link to conductive_interconnect_element_with_pre_defined_transitions (as associated_element)	PATH			conductive_interconnect_element_terminal_link <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition product_definition characterized_product_definition

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition product_definition {[product_definition.description = 'laminated component'] [product_definition => component_definition]} characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => {shape_aspect (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting')} component_shape_aspect
CONNECTION_ZONE_ BASED_FABRICATION_JOINT	connection_zone_based_ fabrication_joint	210		connection_zone_based_fabrication_joint<= fabrication_joint <= shape_aspect
connection_zone_based_ fabrication_joint to connection_zone (as zone_1)	PATH			connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 1'}

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 2'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
connection_zone_based_- fabrication_joint to connection_zone (as zone_2)	PATH			
DEPENDENTLY_LOCATED_- LAYER_CONNECTION_POINT	layer_connection_point	210	333	<pre> layer_connection_point <= shape_aspect {shape_aspect shape_aspect.description = 'dependently located'} </pre>
dependently_located_layer_- connection_point to connection_zone (as reference_zone)	PATH			<pre> layer_connection_point <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference zone'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
dependently_located_layer_- connection_point to embedded_component_terminal (as associated_design_object)	PATH			<pre> layer_connection_point <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated design object'} shape_aspect_relationship.relatng_shape_- aspect -> </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal</pre>
dependently_located_layer_- connection_point to inter_stratum_feature (as associated_design_object)	PATH		333	<pre>layer_connection_point <= shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated design object'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature</pre>
dependently_located_layer_- connection_point to interconnect_module_terminal (as associated_design_object)	PATH		340, 265, 266, 263, 264, 268, 267, 333	<pre>layer_connection_point <= shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated design object'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect</pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'interconnect module'} shape_aspect => interconnect_module_terminal layer_connection_point <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'associated design object'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal </pre>
dependently_located_layer_- connection_point to printed_component_join_terminal (as associated_design_object)	PATH			
DIELECTRIC_CROSSOVER_- AREA	stratum_concept_relationship	210	344	<pre> stratum_concept_relationship <= [shape_aspect] [shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'dielectric crossover area'}] </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
dielectric_crossover_area to stratum_feature (as related_concept)	PATH		344, 333	stratum_concept_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature
dielectric_crossover_area to stratum_feature (as relating_concept)	PATH			stratum_concept_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature
dielectric_crossover_area to stratum_feature (as resulting_dielectric_crossover)	PATH		333, 344	stratum_concept_relationship <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'resulting dielectric crossover'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature
EXPLICITLY LOCATED - LAYER_CONNECTION_POINT	layer_connection_point	210	333, 167	layer_connection_point <= shape_aspect {shape_aspect shape_aspect.description = 'explicitly located'}
FABRICATION_JOINT	fabrication_joint	210		fabrication_joint <= [shape_aspect] [shape_aspect_relationship] fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_1)	PATH			

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination_passage join terminal'} shape_aspect => component_terminal
fabrication_joint to conductive_- interconnect_element_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} shape_aspect => component_terminal
fabrication_joint to conductive_- interconnect_element_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} shape_aspect => component_terminal
fabrication_joint to interconnect_module (as associated_interconnect_module)	PATH			fabrication_joint <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition {product_definition.name = 'interconnect module'}
fabrication_joint to land_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
fabrication_joint to land_join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
fabrication_joint to non_- functional_land_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fabrication_joint to non_-functional_land_join_terminal (as fabrication_features_2)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to printed_component_join_terminal (as fabrication_features_1)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to printed_component_join_terminal (as fabrication_features_2)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to stratum_-feature_template_component (as auxiliary_join_material)	PATH		340, 338, 268, 333	<pre> fabrication_joint <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'auxiliary join material'} </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> shape_aspect {[shape_aspect.description = 'stratum feature template component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'lamine component']] shape_aspect => component_shape_aspect </pre>
fabrication_joint to via_terminal (as fabrication_features_1)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal </pre>
fabrication_joint to via_terminal (as fabrication_features_2)	PATH			<pre> fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal
INTER_STRATUM_JOIN_IMPLEMENTATION	(stratum_concept_relationship) (plated_inter_stratum_feature)	210 210	333	(stratum_concept_relationship <= [shape_aspect {shape_aspect shape_aspect.description = 'physical net supporting stratum feature conductive join'}] [shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature conductive join'}]) (plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'plated cutout') (shape_aspect.description = 'plated cutout edge segment') (shape_aspect.description = 'plated interconnect module edge segment') (shape_aspect.description = 'plated interconnect module edge')})</pre>
inter_stratum_join_- implementation to inter_stratum_join_relationship (as topological_requirement)	PATH			<pre>(stratum_concept_relationship <=) (plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'join implementation'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.name = 'inter stratum join'} shape_aspect => join_shape_aspect</pre>
INTER_STRATUM_JOIN_RE- LATIONSHIP	join_shape_aspect	210	333	<pre>join_shape_aspect <= shape_aspect {shape_aspect shape_aspect.name = 'inter stratum join'}</pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
inter_stratum_join_relationship to layer_connection_point (as points_to_be_connected)	PATH			<pre> join_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'connected point'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point </pre>
INTRA_STRATUM_JOIN_RE- LATIONSHIP	join_shape_aspect	210	333	<pre> join_shape_aspect <= shape_aspect {shape_aspect shape_aspect.name = 'intra stratum join'} </pre>
intra_stratum_join_relationship to layer_connection_point (as points_to_be_connected)	PATH			<pre> join_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'connected point'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point </pre>
JOIN_RELATIONSHIP	join_shape_aspect	210		<pre> join_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.name = 'unrouted join') (shape_aspect.name = 'constrained intra layer join') (shape_aspect.name = 'inter stratum join') (shape_aspect.name = 'intra stratum join')} </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
join_relationship to interconnect_module (as associated_layout)	PATH			<pre> join_shape_aspect <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {product_definition.name = 'interconnect module'} </pre>
JUNCTION_VERTEX_ALLOCATION	shape_aspect_relationship	41	198, 199	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'junction vertex_allocation'} </pre>
junction_vertex_allocation to explicitly_located_layer_-connection_point (as implementation_point)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'explicitly located'} shape_aspect => layer_connection_point </pre>
junction_vertex_allocation to inter_stratum_join_relationship (as implementation_point)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.name = 'inter stratum join'} shape_aspect => join_shape_aspect </pre>
junction_vertex_allocation to topological_junction (as required_vertex)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'topological junction'} </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
LAYER_CONNECTION_POINT	layer_connection_point	210	333	layer_connection_point <= shape_aspect
name	shape_aspect.name	41		layer_connection_point <= shape_aspect shape_aspect.name
layer_connection_point to cartesian_point (as location_2d)	PATH			layer_connection_point <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {[representation => shape_representation] [representation representation.name = 'connection point location 2d']} representation representation.items[i] -> representation_item => geometric_representation_item => {geometric_representation_item geometric_representation_item.dim = 2} point => cartesian_point
layer_connection_point to cartesian_point (as location_3d)	PATH			layer_connection_point <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {[representation => shape_representation] [representation representation.name = 'connection point location 3d']}] representation representation.items[i] -> representation_item => geometric_representation_item {geometric_representation_item geometric_representation_item.dim = 3} point => cartesian_point
layer_connection_point to design_layer_stratum (as resident_design_layer_stratum)	PATH		340, 265, 266, 338, 263, 264, 268, 267	layer_connection_point <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.name = 'design layer'} product_definition => stratum
LAYER_CONNECTION_- POINT_LINK	layer_connection_point_link	210		{[layer_connection_point_link <= shape_aspect] [layer_connection_point_link <= shape_aspect_relationship]] layer_connection_point_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect => layer_connection_point layer_connection_point_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point layer_connection_point_link <= shape_aspect
layer_connection_point_link to layer_connection_point (as precedent_point)	PATH			product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition = product_ definition
layer_connection_point_link to layer_connection_point (as subsequent_point)	PATH			
layer_connection_point_link to stratum (as associated_stratum)	PATH			

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => stratum
PARTIALLY_ROUTED_PHYSICAL_NETWORK	physical_network	210		physical_network <= shape_aspect {shape_aspect.description = 'partially routed physical network'}
partially_routed_physical_network to unrouted_join_relationship (as unrouted_connectivity_ requirement_element)	PATH			physical_network <= shape_aspect <= shape_aspect.relationship.relatng_shape_ aspect shape_aspect.relationship {shape_aspect.relationship connectivity_requirement'} shape_aspect.relationship.related_shape_aspect -> shape_aspect => {shape_aspect shape_aspect.name = 'unrouted join'} join_shape_aspect
PHYSICAL_COMPONENT_ TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') (shape_aspect.description = 'packaged component join terminal'))})

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(component_interface_terminal <= shape_aspect)
PHYSICAL_CONNECTIVITY_- ABSTRACTION_MAP	shape_aspect_relationship	41	239	{shape_aspect_relationship shape_aspect_relationship.name = 'physical to topological abstraction map'}
physical_connectivity_- abstraction_map to conductive_interconnect_element (as geometric_context_based_- definition)	PATH		333, 340, 338	shape_aspect_relationship aspect -> shape_aspect {shape_aspect [(shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => component_definition] [shape_aspect => component_shape_aspect]}}
physical_connectivity_- abstraction_map to physical_connectivity_element (as name_space_context_based_- definition)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_element

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_CONNECTIVITY_- DEFINITION	physical_connectivity_definition	210	333	physical_connectivity_definition <= shape_aspect
name	shape_aspect.name	41	333	physical_connectivity_definition <= shape_aspect shape_aspect.name
physical_connectivity_definition to physical_unit_network_definition (as associated_definition)	PATH			physical_connectivity_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => physical_unit_network_definition
physical_connectivity_definition to physical_component_terminal (as associated_terminals)	PATH			physical_connectivity_definition <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated terminals'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'interconnect component join terminal')}

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'packaged component join terminal') } shape_aspect => component_terminal</pre>
PHYSICAL_CONNECTIVITY_- DEFINITION_RELATIONSHIP	shape_aspect_relationship	41	240	<pre>{shape_aspect_relationship shape_aspect_relationship.name = 'physical connectivity definition change'}</pre>
physical_connectivity_definition_- relationship to ee_requirement_occurrence (as requirement)	PATH			<pre>shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'requirement'} property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'requirement'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property</pre>
physical_connectivity_definition_- relationship to physical_connectivity_definition (as current_definition)	PATH		333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition</pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_connectivity_definition_- relationship to physical_connectivity_definition (as previous_definition)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition
PHYSICAL_CONNECTIVITY_- ELEMENT	physical_connectivity_element	210	241	physical_connectivity_element <= [shape_aspect] [shape_aspect_relationship]
name	shape_aspect.name	41	333	physical_connectivity_element <= shape_aspect shape_aspect.name
physical_connectivity_element to component terminal (as end_terminus)	PATH		344, 333	physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'conductive interconnect element terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'packaged component join terminal')}

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'printed component join terminal') (shape_aspect.description = 'via terminal') shape_aspect => component_terminal</pre>
physical_connectivity_element to component_terminal (as start_terminus)	PATH		344, 333	<pre>physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'}} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'assembly module component terminal') (shape_aspect.description = 'bare die component terminal') (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'conductive interconnect element terminal') (shape_aspect.description = 'interconnect component join terminal') (shape_aspect.description = 'interconnect module component terminal') (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'packaged component join terminal') (shape_aspect.description = 'minimally defined component terminal') (shape_aspect.description = 'printed component join terminal') (shape_aspect.description = 'via terminal') shape_aspect => component_terminal</pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_connectivity_element to physical_connectivity_structure_- definition (as connectivity_context)	PATH			<pre> physical_connectivity_element <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'structure element'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition </pre>
physical_connectivity_element to topological_junction (as end_terminus)	PATH		344	<pre> physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'topological junction'} </pre>
physical_connectivity_element to topological_junction (as start_terminus)	PATH			<pre> physical_connectivity_element <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start to end terminus'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'topological junction'} </pre>
PHYSICAL_CONNECTIVITY_- STRUCTURE_DEFINITION	physical_connectivity_definition	210	333	<pre> physical_connectivity_definition <= shape_aspect </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_NETWORK	physical_network	210		physical_network <= shape_aspect
name	shape_aspect.name	41		physical_network <= shape_aspect shape_aspect.name
physical_network to interconnect_module (as current_associated_definition)	PATH			physical_network <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) physical_network <= shape_aspect <= shape_aspect.relationship.related_shape_aspect {shape_aspect_relationship
physical_network to interconnect_module (as prior_associated_definition)	PATH			

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'prior associated_definition' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) physical_network <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship </pre>
physical_network to join_relationship (as interconnect_module - network_topology_element)	PATH			<pre> physical_network <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'network topology' shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.name = 'constrained intra layer join') (shape_aspect.name = 'inter stratum join') (shape_aspect.name = 'intra stratum join')} shape_aspect => join_shape_aspect physical_network <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'connectivity requirement'}} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'printed component join terminal')} shape_aspect => component_terminal physical_network <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship </pre>
physical_network to laminate_- component_join_terminal (as connectivity_requirement_- element)	PATH			
physical_network to part_- connected_terminals_definition (as reference_connected_terminals)	PATH			

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'reference connected terminals' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_connected_terminals_definition </pre>
PHYSICAL_NETWORK_SUP- PORTING_INTER_STRATUM_- FEATURE	plated_inter_stratum_feature	210		<pre> plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {[(shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => (plated_interconnect_module_edge_segment) (plated_cutout_edge_segment)) (shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'join two physical connectivity definition supporting inter stratum feature') (shape_aspect.description = 'plated conductive base blind via') (shape_aspect.description = 'plated interconnect module edge') (shape_aspect.description = 'plated cutout'))] [shape_aspect <- shape_aspect_relationship.related_shape_aspect </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship { shape_aspect_relationship.name = 'join implementation' } shape_aspect_relationship.relatng_shape_- aspect -> { shape_aspect shape_aspect.name = 'inter stratum join' } shape_aspect => join_shape_aspect[] </pre>
PHYSICAL_NETWORK_SUPPORTING_STRATUM_FEATURE_CONDUCTIVE_JOIN	stratum_concept_relationship	210		<pre> stratum_concept_relationship <= [shape_aspect { shape_aspect shape_aspect.description = 'physical net supporting stratum feature conductive join' } [shape_aspect_relationship { shape_aspect_relationship.name = 'stratum feature conductive join' }]] </pre>
PHYSICAL_UNIT_NETWORK_DEFINITION	physical_unit_network_definition	210		<pre> physical_unit_network_definition <= product_definition { product_definition product_definition.formatation -> product_definition.formatation product_definition.formatation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'physical network design' } </pre>
STRATUM_CONCEPT_RELATIONSHIP	stratum_concept_relationship	210	344	<pre> stratum_concept_relationship <= [shape_aspect [shape_aspect_relationship </pre>

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect_relationship shape_aspect_relationship.name = 'stratum concept relationship'}}
stratum_concept_relationship to ee_requirement_occurrence (as technology_requirement)	PATH		333	stratum_concept_relationship <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.name = 'technology requirement'} property_definition <= property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'technology_requirement'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property
stratum_concept_relationship to shape_definition (as resulting_shape)	PATH			stratum_concept_relationship <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
STRUTUM_FEATURE_CONDUCTIVE_JOIN	stratum_concept_relationship	210	344	stratum_concept_relationship <=[shape_aspect] {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature conductive join'}}
stratum_feature_conductive_join to stratum_feature_template_ component (as related_concept)	PATH		344, 333	stratum_concept_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'stratum feature template component'} shape_aspect => component_shape_aspect
stratum_feature_conductive_join to stratum_feature_template_ component (as relating_concept)	PATH			stratum_concept_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> {shape_aspect shape_aspect.description = 'stratum feature template component'} shape_aspect => component_shape_aspect
TOPOLOGICAL_JUNCTION	shape_aspect	41	372, 373	{shape_aspect [shape_aspect.description = 'topological junction'] [shape_aspect.product_definitional = .TRUE.]}
name	shape_aspect.name	41		
topological_junction to physical_ connectivity_structure_definition (as scope)	PATH		333	shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship

Table 59 – Mapping table for interconnect_module_connection_routing UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'topological junction scope'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition </pre>
UNROUTED_JOIN_RELATIONSHIP	join_shape_aspect	210		<pre> join_shape_aspect <= shape_aspect {shape_aspect shape_aspect.name = 'unrouted join'} </pre>
unrouted_join_relationship to laminar_component_join_- terminal (as unrouted_terminals)	PATH			<pre> join_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'unrouted terminals'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'embedded component terminal') (component_terminal) join_terminal'}} component_terminal </pre>

Table 60 – Mapping table for interconnect_module_design UoF

Application element	AIM element	Source	Rules	Reference path
ADJACENT_STRATUM_SURFACE_DEFINITION	shape_aspect_relationship	41	2, 3	shape_aspect_relationship {shape_aspect_relationship.description = 'adjacent stratum surface definition'}
name	shape_aspect_relationship.name	41		
adjacent_stratum_surface_- definition to stratum_surface (as precedent_surface)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'secondary surface'} shape_aspect => stratum_surface
adjacent_stratum_surface_- definition to stratum_surface (as subsequent_surface)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'primary surface'} shape_aspect => stratum_surface
ADJACENT_STRATUM_SURFACE_EMBEDDED_COMPONENT_SURFACE_DEFINITION	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'adjacent stratum surface embedded component surface'}
name	descriptive_representation_- item.description	45		shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition {property_definition property_definition.description = 'adjacent stratum surface embedded component surface characterization'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'adjacent stratum surface embedded component surface name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
shape_element_relationship to stratum_surface (as precedent_item)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => stratum_surface </pre>
shape_element_relationship to stratum_surface (as subsequent_item)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_surface </pre>
shape_element_relationship to component_feature (as precedent_item)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {(shape_aspect => component_terminal) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => component_feature) {(shape_aspect => part_mounting_feature => bare_die_surface => bare_die_top_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_edge_surface) (shape_aspect => bare_die_edge_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_bottom_surface) (shape_aspect => package_body_surface => package_body_top_surface) (shape_aspect => package_body_surface => package_body_edge_surface) (shape_aspect => package_body_edge_surface) (shape_aspect => package_body_surface => package_body_bottom_surface)} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {(shape_aspect => component_terminal) </pre>
shape_element_relationship to component_feature (as subsequent_item)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => component_feature) {(shape_aspect => part_mounting_feature => bare_die_surface => bare_die_top_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_edge_surface) (shape_aspect => bare_die_edge_surface) (shape_aspect => bare_die_edge_segment_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_bottom_surface) (shape_aspect => package_body_surface => package_body_top_surface) (shape_aspect => package_body_surface => package_body_edge_surface) (shape_aspect => package_body_edge_segment_surface) (shape_aspect => package_body_surface => package_body_bottom_surface)}} </pre>
ARRAY_PLACEMENT_GROUP	(array_placement_group - component_shape_aspect)	210	338, 333	<pre> (array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect)</pre>
ASSEMBLY_COMPONENT	(component_definition) (component_shape_aspect)	210 210	333, 340, 338	<pre>(component_definition <= (product_definition {[product_definition product_definition.description != 'printed component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence']}) (product_definition {[product_definition product_definition.description = 'printed component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence']}) (component_shape_aspect <= shape_aspect {(shape_aspect => component_shape_aspect =></pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (inter_stratum_feature) (land) (material_removal_component_shape_aspect) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (assembly_group_component_shape_aspect => footprint_occurrence) (assembly_group_component_shape_aspect => padstack_occurrence) (integral_shield)) (shape_aspect (shape_aspect.description = 'laminare component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'laminare text string component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminare component') (shape_aspect.description = 'stratum feature template component'))} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition product_definition.description = 'laminated component'] [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence'] [product_definition product_definition.description = 'laminated component']] component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'usage definition'} shape_aspect_relationship.related_shape_ aspect -> shape_aspect.of_shape -> </pre>
assembly_component to footprint_definition (as usage_definition)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape <= property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => footprint_definition
assembly_component to padstack_definition (as usage_definition)	PATH			component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship definition'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => padstack_definition
assembly_component to part_template (as usage_definition)	PATH		333	component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => part_template_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_component to physical_unit (as usage_definition)	PATH		338	<pre> component_definition <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship 'instantiated part'} product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) (physical_unit => externally_defined_physical_unit) </pre>
ASSEMBLY_COMPOSITION_- RELATIONSHIP	assembly_component_usage	44	354, 339, 15	<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly composition'} </pre>
assembly_composition_- relationship to assembly_component (as component)	PATH		354, 339, 338, 340, 333	<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition => component_definition} (product_definition [product_definition => component_definition] [product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect]) </pre>
assembly_composition_- relationship to ee_material (as assembly_bonding_agent)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- material_designation.definitions[i] material_designation </pre>
assembly_composition_- relationship to physical_unit_design_view (as assembly)	PATH		354, 339, 338	<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design' product_definition => physical_unit assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
assembly_composition_- relationship to security_classification (as security_code)	PATH			
ASSEMBLY_GROUP_COMPO- NENT	(assembly_group_component_- definition) (assembly_group_component_- shape_aspect)	210 210	338, 333	<pre> (assembly_group_component_definition <= component_definition <= product_definition {product_definition <- product_definition_relationship.relatiing_- product_definition {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship product_definition_relationship.related_- product_definition -> [product_definition => component_definition] [product_definition product_definition.frame_of_reference -> product_definition_context <= </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'physical occurrence' } (assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect {[shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect] [shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => component_definition => assembly_group_component_definition]]) </pre>
BLIND_VIA	plated_passage	210	333	<pre> plated_passage <= plated_interstratum_feature <= interstratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')}
BONDED_CONDUCTIVE_- BASE_BLIND_VIA	plated_passage	210	333	plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'bonded conductive base blind via'}
bonded_conductive_base_blind_- via to stratum_feature_conductive_join (as features_join)	PATH		333	plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'features join'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_concept_relationship {stratum_concept_relationship <= shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature conductive join'}
BURIED_VIA	plated_passage	210		plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'buried via'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COMPONENT_BASED_AN- NOTATION_SYMBOL_PLACE- MENT_RELATIONSHIP	shape_aspect_relationship	41	333, 340, 265, 266, 338	<pre> shape_aspect_relationship { shape_aspect_relationship.name = 'component based_annotation_symbol_placement' } shape_aspect_relationship.related_shape_- aspect -> (shape_aspect => component_shape_aspect) (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition) </pre>
component_based_annotation_- symbol_placement_relationship to special_symbol_laminate_- component (as related_stratum_concept)	PATH		333, 340, 265, 266, 338, 263, 264, 268, 267	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect { [shape_aspect.description = 'special symbol laminate component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']}] shape_aspect => component_shape_aspect
COMPONENT_BASED_ANNO- TATION_TEXT_RELATION- SHIP	shape_aspect_relationship	41	333, 340, 265, 266, 338	shape_aspect_relationship {shape_aspect_relationship.name = 'component based annotation text'} shape_aspect_relationship.relatng_shape_ aspect -> (shape_aspect => component_shape_aspect) (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition)
message_type	descriptive_representation_item	45		shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'message type'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'component reference designation') (descriptive_representation_item.description = 'terminal designation')}} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect [shape_aspect.description = 'laminated text string component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition </pre>
component_based_annotation_- text_relationship to laminated_text_string_component (as related_stratum_concept)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] shape_aspect => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect </pre>
COMPONENT_EXTERNAL_ REFERENCE	representation	43	33	<pre> {representation representation.name = 'component external reference'} </pre>
design_owner	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'design owner'} representation_item => descriptive_representation_item </pre>
part_number	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item representation_item.name = 'part number'} representation_item => descriptive_representation_item </pre>
revision_code	descriptive_representation_item	45		<pre> representation representation.items[i] -> {representation_item </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item.name = 'revision code'} representation_item => descriptive_representation_item
product_definition_id	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'product definition id'} representation_item => descriptive_representation_item
reference_designation	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'reference designation'} representation_item => descriptive_representation_item
COMPONENT_FEATURE	(component_terminal) (component_interface_terminal) (component_feature) (laminar_component_interface_ - terminal) (interconnect_module_ - component_surface_feature)	210 210 210 210 210	333, 37	(component_terminal <= shape_aspect {shape_aspect [shape_aspect.product_definitional = .TRUE.] [(shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'conductive interconnect element terminal') (shape_aspect.description = 'interconnect module component stratum based terminal')])

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'land join terminal') (shape_aspect.description = 'minimally defined component terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'printed component join terminal') (shape_aspect.description = 'via terminal')}]}) (component_interface_terminal <= shape_aspect) (component_feature <= shape_aspect) (laminate_component_interface_terminal <= shape_aspect) (interconnect_module_component_surface_feature <= shape_aspect)</pre>
component_feature to assembly_component (as associated_component)	PATH		340, 338, 333	<pre>((component_terminal <=) (component_interface_terminal <=) (component_feature <=) (laminate_component_interface_terminal <=) (interconnect_module_component_surface_feature <=) shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => component_definition) ((component_terminal <=) (laminate_component_interface_terminal <=) shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminate component']} [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_feature to component_termination_passage_ template_terminal (as definition)	PATH		333	(component_terminal <=) (laminatate_component_interface_terminal <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}
component_feature to land_template_terminal (as definition)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect {shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal')}} (component_terminal <=) (laminatate_component_interface_terminal <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}
component_feature to part_feature (as definition)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect => land_template_terminal (component_terminal <=) (component_interface_terminal <=) (component_feature <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']] (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature)

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
component feature to printed_part_template_terminal (as definition)	PATH			(component_terminal <= laminate_component_interface_terminal <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect => printed_part_template_terminal
component feature to via_template_terminal (as definition)	PATH		333	(component_terminal <= laminate_component_interface_terminal <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'via template terminal'} </pre>
COMPONENT_FEATURE_EXTERNAL_REFERENCE	descriptive_representation_item	45	34, 35	<pre> descriptive_representation_item <= {representation_item representation_item.name = 'component feature external reference'} representation_item </pre>
name	descriptive_representation_item.description	45		
component_feature_external_reference to component_external_reference (as associated_component)	PATH			<pre> descriptive_representation_item <= representation_item <- representation.items[i] {representation representation.name = 'component external reference'} representation </pre>
COMPONENT_FEATURE_MAKE_FROM_RELATIONSHIP	make_from_feature_relationship	210		<pre> make_from_feature_relationship <= shape_aspect_relationship </pre>
component_feature_make_from_relationship to component_feature (as resultant_feature)	PATH		344, 333	<pre> make_from_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_feature_make_from_- relationship to component_feature (as reusable_feature)	PATH			(laminated_component_interface_terminal) (component_feature)) make_from_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.relation_shape_- aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
COMPONENT_FEATURE_RELATIONSHIP	component_feature_relationship	210		{[component_feature_relationship <= shape_aspect] [component_feature_relationship <= shape_aspect_relationship]}
component_feature_relationship to component_feature (as related_component_feature)	PATH		333	component_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
component_feature_relationship to component_feature (as relating_component_feature)	PATH		333	component_feature_relationship <= shape_aspect_relationship shape_aspect_relationship.relation_shape_- aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
COMPONENT_FEATURE_TO_LAYOUT_FEATURE_RELATIONSHIP	shape_aspect_relationship	41		shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'component feature to layout feature'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_feature_to_layout_- feature_relationship to assembly_component (as relating_feature)	PATH		36, 333, 340, 265, 266, 338	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => component_shape_aspect
component_feature_to_layout_- feature_relationship to component_feature (as relating_feature)	PATH		36, 333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => (component_feature) (component_interface_terminal) (laminar_component_interface_terminal) (component_terminal))
component_feature_to_layout_- feature_relationship to ee_requirement_occurrence (as requirement)	PATH		340	shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition property_definition.definition <- {property_definition property_definition.name = 'requirement'} property_definition <- property_definition.relationship.related_- property_definition {property_definition.relationship property_definition.relationship.name = 'requirement'} property_definition.relationship.relatng_- property_definition -> property_definition => requirements_property
component_feature_to_layout_- feature_relationship to inter_stratum_feature (as related_stratum_concept)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_feature_to_layout_- feature_relationship to laminated_component (as related_stratum_concept)	PATH		36, 333, 340, 263, 264, 268, 267, 265, 266, 338	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminated text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect {(shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component') (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => {product_definition.description = 'laminated component'} component_definition) </pre>
component_feature_to_layout_- feature_relationship to stratum (as related_stratum_concept)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape product_definition_shape <= </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => stratum
component_feature_to_layout_ feature_relationship to stratum_feature (as related_stratum_concept)	PATH		36	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature
component_feature_to_layout_ feature_relationship to stratum_surface (as related_stratum_concept)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_surface
COMPONENT_INTERFACE_ TERMINAL	component_interface_terminal	210	333	component_interface_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect component interface terminal') (shape_aspect.description = 'packaged connector component interface terminal')}}
COMPONENT_MAKE_FROM_ RELATIONSHIP	(make_from_usage_option) (laminare_group_component_ make_from_relationship)	44 210		(make_from_usage_option) (laminare_group_component_make_from_ relationship <= make_from_usage_option)
component_make_from_ relationship to assembly_component (as resultant_component)	PATH			(make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition =>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> component_definition (laminate_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => assembly_group_component_definition) </pre>
<pre> component_make_from_- relationship to assembly_component (as reusable_component) </pre>	PATH			<pre> (make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition) (laminate_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => assembly_group_component_definition) </pre>
COMPONENT_TERMINAL	<pre> (component_terminal) (component_interface_terminal) (component_feature) (laminate_component_interface_- terminal) </pre>	210 210 210 210	333	<pre> (component_terminal <= shape_aspect {shape_aspect [shape_aspect.product_definitional = .TRUE.] [(shape_aspect.description = 'component termination passage join terminal') </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'printed component join terminal')}]}) (component_interface_terminal <= shape_aspect) (component_feature <= shape_aspect) (laminate_component_interface_terminal <= shape_aspect)</pre>
global_swappable	descriptive_representation_item	45		<pre>component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'global swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false') (descriptive_representation_item.description = 'true')} descriptive_representation_item</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
local_swappable	descriptive_representation_item	45		<pre> component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'local swappable'} representation_item => {descriptive_representation_item (descriptive_representation_item.description = 'false')} (descriptive_representation_item.description = 'true')}} descriptive_representation_item </pre>
swap_code	descriptive_representation_ - item.description	45		<pre> component_terminal <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.items[i] -> { representation_item { representation_item.name = 'swap code' } representation_item => { descriptive_representation_item { descriptive_representation_item.description } } } } </pre>
COMPONENT_TERMINATION_PASSAGE	plated_passage	210	333	<pre> plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= { shape_aspect { shape_aspect.description = 'component termination passage' } } </pre>
component_termination_passage to component_termination_ passage_template (as usage_definition)	PATH		333, 340	<pre> plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= { shape_aspect <- { shape_aspect.relationship.related_shape_aspect { shape_aspect_relationship { shape_aspect_relationship.name = 'instantiated template' } } } shape_aspect_relationship.related_shape_ aspect -> { shape_aspect { shape_aspect.description = 'component termination passage template' } } shape_aspect => { part_template_definition } } </pre>
COMPONENT_TERMINATION_PASSAGE_INTERFACE_TERMINAL	lamineate_component_interface_ terminal	210	333	<pre> lamineate_component_interface_terminal <= { shape_aspect { shape_aspect { shape_aspect.description = 'component termination passage interface terminal' } } } </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_termination_passage_- interface_terminal to component_termination_passage (as associated_component)	PATH		333	<pre> lamineate_component_interface_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'lamineate component'] [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_termination_passage_- interface_terminal to component_termination_passage_- template_interface_terminal (as definition)	PATH			laminatE_component_interface_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}} shape_aspect_relationship.relatinG_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage template interface terminal'}} shape_aspect
COMPONENT_TERMINA- TION_PASSAGE_JOIN_TERMINAL	component_terminal	210	333	component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'component termination passage join terminal'}
component_termination_passage_- join_terminal to component_termination_passage (as associated_component)	PATH		333	component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminatE component']}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition]]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship.relatng_shape_- aspect -></pre>
<pre>component_termination_passage_- join_terminal to component_termination_passage_- template_join_terminal (as definition)</pre>	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect shape_aspect.description = 'component termination passage template join terminal'} shape_aspect
CONDUCTIVE_FILLED_AREA	stratum_feature	210	333	stratum_feature <= shape_aspect {shape_aspect shape_aspect.description = 'conductive filled area'}
CONDUCTIVE_INTERCON- NECT_ELEMENT	component_shape_aspect	210	333, 340, 338	component_shape_aspect <= shape_aspect {shape_aspect [(shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition]}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
conductive_interconnect_element to stratum_feature (as composed_conductor)	PATH			<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed conductor'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature </pre>
CONDUCTIVE_INTERCON- NECT_ELEMENT_TERMINAL	component_terminal	210	333, 57	<pre> component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} </pre>
name	shape_aspect.name	41		<pre> component_terminal <= shape_aspect shape_aspect.name </pre>
conductive_interconnect_- element_terminal to conductive_interconnect_element (as associated_component)	PATH		333, 340, 338	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminated component']} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition]] characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect [(shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition')] [shape_aspect => component_shape_aspect]] component_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -></pre>
conductive_interconnect_ element_terminal_to connection_zone (as connection)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} component_terminal <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect.relationship {shape_aspect.relationship shape_aspect.relationship.name = 'terminal location'} shape_aspect.relationship.relatng_shape_- aspect -> shape_aspect => layer_connection_point
conductive_interconnect_- element_terminal to layer_connection_point (as location)	PATH			
CONDUCTIVE_INTERCON- NECT_ELEMENT_WITH_- PRE_DEFINED_TRANSITIONS	component_shape_aspect	210	58, 333, 340, 338	{component_shape_aspect <= shape_aspect [(shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition => component_definition]}
conductive_interconnect_- element_with_pre_defined_- transitions to curve (as centreline_shape)	PATH			component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition { property_definition { property_definition.name = 'centreline shape' } property_definition <- property_definition_representation.definition property_definition_representation { property_definition_representation.used_ representation -> { representation => { shape_representation { representation.items[i] -> representation_item { representation_item { representation_item.name = 'centreline' } representation_item => geometric_representation_item => curve } } } } } } </pre>
conductive_interconnect_ - element_with_pre_defined_ - transitions to trace_template (as usage_definition)	PATH		333	<pre> component_shape_aspect <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship { shape_aspect_relationship { shape_aspect_relationship.name = 'instantiated template' } shape_aspect_relationship.relatng_shape_ aspect -> { shape_aspect { shape_aspect.description = 'trace template' } } shape_aspect => part_template_definition } </pre>
CONDUCTIVE_INTERCON- NECT_ELEMENT_WITH_- USER_DEFINED_SINGLE_- TRANSITION	component_shape_aspect	210	59, 340, 338	<pre> { component_shape_aspect <= shape_aspect { [shape_aspect.description = 'conductive interconnect element with user defined single transition'] [shape_aspect.of_shape -> product_definition_shape <= property_definition { property_definition.definition -> characterized_definition } } } </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition[]
conductive_interconnect_ element_with_user_defined_ single_transition to layer_connection_point (as end_terminus)	PATH		333	component_shape_aspect <= shape_aspect shape_aspect <- shape_aspect_relationship.relying_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'end terminus'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point
conductive_interconnect_ element_with_user_defined_ single_transition to layer_connection_point (as start_terminus)	PATH			component_shape_aspect <= shape_aspect shape_aspect <- shape_aspect_relationship.relying_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'start terminus'} shape_aspect_relationship.related_shape_aspect ->

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
conductive_interconnect_- element_with_user_defined_- single_transition to bounding_curve (as left_curve_shape_representation)	PATH		333	<pre> shape_aspect => layer_connection_point component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> {representation => shape_representation} representation representation.items[i] -> representation_item => {representation_item representation_item.name = 'left curve shape representation'} geometric_representation_item => curve => bounded_curve => trimmed_curve </pre>
conductive_interconnect_- element_with_user_defined_- single_transition to trimmed_line (as left_curve_shape_representation)	PATH			<pre> component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation => shape_representation} representation.items[i] -> representation_item => {representation_item representation_item.name = 'left curve shape representation'} geometric_representation_item => curve => trimmed_curve </pre>
<pre> conductive_interconnect_ element_with_user_defined_ single_transition to bounding_curve (as right_curve_shape_representation) </pre>	PATH		333	<pre> component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation => shape_representation} representation representation.items[i] -> representation_item => {representation_item representation_item.name = 'right curve shape representation'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_representation_item => curve => bounded_curve => trimmed_curve </pre>
conductive_interconnect_- element_with_user_defined_- single_transition to trimmed_line (as right_curve_shape_representation)	PATH			<pre> component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> {representation => shape_representation} representation representation.items[i] -> representation_item => {representation_item representation_item.name = 'right curve shape representation'} geometric_representation_item => curve => trimmed_curve </pre>
conductive_interconnect_- element_with_user_defined_- single_transition to trace_template (as end_definition)	PATH		333	<pre> component_shape_aspect <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship [shape_aspect_relationship.name = 'instantiated template']} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[shape_aspect_relationship.description = 'end feature'] shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'trace template'} shape_aspect => part_template_definition</pre>
conductive_interconnect_- element_with_user_defined_- single_transition to trace_template (as start_definition)	PATH			<pre>component_shape_aspect <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship [shape_aspect_relationship.name = 'instantiated template']} [shape_aspect_relationship.description = 'start feature']] shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'trace template'} shape_aspect => part_template_definition</pre>
CONDUCTOR	stratum_feature	210		<pre>stratum_feature <= shape_aspect {shape_aspect shape_aspect.description = 'conductor'}</pre>
conductor to intra_stratum_join_relationship (as implementation_requirement)	PATH		333	<pre>stratum_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'join implementation' shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.name = 'intra stratum join'} shape_aspect => join_shape_aspect stratum_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'design layer'} product_definition => stratum </pre>
conductor to stratum (as resident_stratum)	PATH			
CONNECTED_AREA_COMPO- NENT	connected_area_component	210		<pre> connected_area_component <= component_shape_aspect <= {shape_aspect [shape_aspect.description = 'stratum feature template component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminare component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']] </pre>
connected_area_component to layer_connection_point_link (as associated_layer_connection_ - point)	PATH			<pre> connected_area_component <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_ - aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated layer connection point'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point_link </pre>
CONNECTED_FILLED_AREA	stratum_feature	210		<pre> stratum_feature <= shape_aspect {shape_aspect shape_aspect.name = 'connected filled area'} </pre>
connected_filled_area to intra_stratum_join_relationship (as implementation_requirement)	PATH		333	<pre> stratum_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'join implementation'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.name = 'intra stratum join'} shape_aspect => join_shape_aspect </pre>
CONNECTION_ZONE_- BASED_FABRICATION_JOINT	connection_zone_based_- fabrication_joint	210		<pre> connection_zone_based_fabrication_joint<= fabrication_joint <= shape_aspect </pre>
connection_zone_based_- fabrication_joint to connection_zone (as zone_1)	PATH			<pre> connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 1'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
connection_zone_based_- fabrication_joint to connection_zone (as zone_2)	PATH			<pre> connection_zone_based_fabrication_joint <= fabrication_joint <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'zone 2'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'connection zone'} </pre>
CUTOUT	inter_stratum_feature	210		<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'cutout')} (shape_aspect.description = 'partially plated cutout') </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'plated cutout') (shape_aspect.description = 'physical connectivity interrupting cutout')
CUTOUT_EDGE_SEGMENT	cutout_edge_segment	210	333	[cutout_edge_segment <= inter_stratum_feature] [cutout_edge_segment <= shape_aspect_relationship]
cutout_edge_segment to cutout (as composed_cutout)	PATH		333	cutout_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed cutout'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature
cutout_edge_segment to edge_segment_vertex (as end_vertex)	PATH			cutout_edge_segment <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
cutout_edge_segment to edge_segment_vertex (as start_vertex)	PATH			<pre> cutout_edge_segment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
cutout_edge_segment to inter_stratum_feature_edge_- segment_template (as usage_definition)	PATH			<pre> cutout_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'inter stratum feature edge segment template'} shape_aspect => part_template_definition => inter_stratum_feature_edge_segment_template </pre>
DESIGN_LAYER_STRATUM	stratum	210	265, 266, 338, 263, 264, 268, 267	<pre> stratum <= product_definition {product_definition product_definition.name = 'design layer'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
primary_design_layer_stratum	product_definition.description	41	265, 266, 338	stratum <= product_definition product_definition.description {(product_definition.description = 'primary design layer stratum') (product_definition.description = 'non primary design layer stratum')}}
design_layer_stratum to design_layer_technology (as of_technology)	PATH		338	stratum <= product_definition characterized_product_definition = product_- definition characterized_product_definition product_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'}} property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object {characterized_object characterized_object.description = 'design layer'} characterized_object => stratum_technology

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DIELECTRIC_MATERIAL_PAS-SAGE	inter_stratum_feature	210	333	inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'dielectric material_passage'}
dielectric_material_passage to cutout (as precedent_passage)	PATH		333	inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'precedent passage'}
dielectric_material_passage to ee_material (as feature_material)	PATH			shape_aspect.relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature
dielectric_material_passage to ee_requirement_occurrence (as feature_material)	PATH			inter_stratum_feature <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= material_designation.definitions[i] material_designation
			333	inter_stratum_feature <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition {property_definition property_definition.name = 'feature material'} property_definition <- property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'feature material'} property_definition.relationship.related_ property_definition -> property_definition => requirements_property </pre>
DOCUMENTATION_LAYER_- STRATUM	stratum	210	263, 264, 268, 267	<pre> stratum <= product_definition {product_definition product_definition.name = 'documentation layer'} </pre>
documentation_layer_stratum to documentation_layer_technology (as of_technology)	PATH		265, 266, 338	<pre> stratum <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'technology usage'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_relationship property_definition_relationship.relatiing_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- object {characterized_object characterized_object.description = 'documentation layer'} characterized_object => stratum_technology </pre>
EDGE_SEGMENT_VERTEX	edge_segment_vertex	210	146, 147	<pre> edge_segment_vertex <= physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'single datum'} </pre>
name	descriptive_representation_- item.description	45		<pre> shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.description = 'shape element characterization'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.name = 'shape element name' representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
ELECTRICAL_ISOLATION_- LAMINATE_COMPONENT	electrical_isolation_component_- shape_aspect	210		<pre> electrical_isolation_component_shape_aspect <= filled_area_material_removal_component_shape_- aspect </pre>
EMBEDDED_COMPONENT_- TERMINAL	component_terminal	210		<pre> component_terminal <= shape_aspect { shape_aspect shape_aspect.description = 'embedded component terminal' } </pre>
embedded_component_terminal to bare_die_component (as associated_component)	PATH			<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition { product_definition product_definition.description = 'bare die component' } product_definition => component_definition component_definition </pre>
embedded_component_terminal to bare_die_terminal (as definition)	PATH			<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship { shape_aspect_relationship </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.name = 'instantiated feature' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal
FABRICATION_JOINT	fabrication_joint	210		fabrication_joint <= [shape_aspect] [shape_aspect_relationship]
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal
fabrication_joint to component_termination_passage_- join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal
fabrication_joint to conductive_- interconnect_element_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
fabrication_joint to conductive_- interconnect_element_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} shape_aspect => component_terminal
fabrication_joint to interconnect_module (as associated_interconnect_module)	PATH			fabrication_joint <= shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {product_definition.name = 'interconnect module'}
fabrication_joint to land_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
fabrication_joint to land_join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'land join terminal'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
fabrication_joint to non- functional_land_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal' } shape_aspect => component_terminal
fabrication_joint to non- functional_land_join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal' } shape_aspect => component_terminal
fabrication_joint to printed_component_join_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal' } shape_aspect => component_terminal
fabrication_joint to printed_component_join_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal' }

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_terminal
fabrication_joint to stratum_ feature_template_component (as auxiliary_joint_material)	PATH		340, 338, 268, 333	fabrication_joint <= shape_aspect <- shape_aspect_relationship.relatng_shape_ aspect shape_aspect_relationship {shape_aspect_relationship.name = 'auxiliary joint_material'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {[shape_aspect.description = 'stratum feature template component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']}] shape_aspect => component_shape_aspect
fabrication_joint to via_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
fabrication_joint to via_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_1)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal
fabrication_joint to embedded_component_terminal (as fabrication_features_2)	PATH			fabrication_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal
FIDUCIAL	fiducial	210	333, 340, 268	fiducial <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <=

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition [product_definition.description = 'laminat component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence']]
FIDUCIAL_STRATUM_FEA-TURE	fiducial_stratum_feature	210		fiducial_stratum_feature <= stratum_feature <= shape_aspect
FILLED_AREA_MATERIAL_REMOVAL_LAMINATE_COMPONENT	filled_area_material_removal_ component_shape_aspect	210		filled_area_material_removal_component_shape_ aspect <= material_removal_component_shape_aspect <= component_shape_aspect
filled_area_material_removal_ laminat_component to padstack_definition (as reference_padstack)	PATH			filled_area_material_removal_component_shape_ aspect <= material_removal_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <=

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => padstack_definition
FOOTPRINT_OCCURRENCE	footprint_occurrence	210		footprint_occurrence <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect footprint_occurrence <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'usage definition'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => footprint_definition
footprint_occurrence to footprint_definition (as usage_definition)	PATH			
FOOTPRINT_OCCURRENCE_- SUB_ASSEMBLY_RELATION- SHIP	footprint_occurrence_shape_- aspect_relationship	210		footprint_occurrence_shape_aspect_relationship <= shape_aspect_relationship

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
sub_assembly_reference_designa- tion	PATH			footprint_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.name footprint_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.description footprint_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect footprint_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => assembly_group_component_shape_aspect <= component_shape_aspect => footprint_occurrence
terminal_location	PATH			
footprint_occurrence_sub_- assembly_relationship to laminare_component (as related_component)	PATH			
footprint_occurrence_sub_- assembly_relationship to footprint_occurrence (as relating_component)	PATH			
FROZEN_ASSIGNMENT	frozen_assignment	210	13	frozen_assignment <= approval_assignment
frozen_assignment to ee_approval (as assigned_approval)	PATH		10, 11	frozen_assignment <= approval_assignment approval_assignment.assigned_approval -> approval {approval approval.status -> approval.status (approval.status.name = 'approved') (approval.status.name = 'not yet approved')}}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
frozen_assignment to frozen_design_object (as frozen_item)	PATH			frozen_assignment frozen_assignment.items[i] -> frozen_assigned_item
INTER_STRATUM_EXTENT	product_definition_relationship	41	188	{product_definition_relationship product_definition_relationship.name = 'inter stratum extent'}
inter_stratum_extent to stratum (as from_stratum)	PATH		265, 266, 338	product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition => stratum
inter_stratum_extent to stratum (as to_stratum)	PATH			product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => stratum
INTER_STRATUM_FEATURE	inter_stratum_feature	210	340, 338, 268	inter_stratum_feature <= component_shape_aspect <= shape_aspect {[(shape_aspect => component_shape_aspect => inter_stratum_feature => (cutout_edge_segment) (interconnect_module_edge_segment) (plated_inter_stratum_feature)) (shape_aspect (shape_aspect.description = 'partially plated cutout') (shape_aspect.description = 'partially plated interconnect module edge') (shape_aspect.description = 'unsupported passage') (shape_aspect.description = 'cutout') (shape_aspect.description = 'physical connectivity interrupting cutout')]

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'dielectric material passage') (shape_aspect.description = 'interconnect module edge'))] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition product_definition.description = 'laminated component'] [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence']]}</pre>
feature_of_size	descriptive_representation_- item.description	45	333	<pre>inter_stratum_feature <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'feature of size'} representation_item => descriptive_representation_item descriptive_representation_item.description {descriptive_representation_item.description (descriptive_representation_item.description = 'true')} (descriptive_representation_item.description = 'false')} </pre>
inter_stratum_feature to inter_stratum_feature_template (as usage_definition)	PATH		333	<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'via template') (shape_aspect.description = 'component termination passage template') (shape_aspect.description = 'unsupported passage template') (shape_aspect.description = 'inter stratum feature edge template')} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'counterbore passage template') (shape_aspect.description = 'countersunk passage template') (shape_aspect.description = 'blind passage template')}} shape_aspect => part_template_definition</pre>
INTER_STRATUM_FEATURE_- EDGE_SEGMENT_TEMPLATE	inter_stratum_feature_edge_- segment_template	210		<pre>[inter_stratum_feature_edge_segment_template <= part_template_definition] [inter_stratum_feature_edge_segment_template <= shape_aspect_relationship]</pre>
inter_stratum_feature_edge_- segment_template to edge_segment_cross_section (as cross_section)	PATH			<pre>inter_stratum_feature_edge_segment_template <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- {property_definition property_definition.name = 'cross section'} property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation => volume_shape_intersection => edge_segment_cross_section</pre>
inter_stratum_feature_edge_- segment_template to inter_stratum_feature_edge_- template (as composed_edge)	PATH			<pre>inter_stratum_feature_edge_segment_template <= part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship.related_shape_- aspect -> shape_aspect => part_template_definition => inter_stratum_feature_edge_template</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
inter_stratum_feature_edge_- segment_template to edge_segment_vertex (as end_vertex)	PATH			<pre> inter_stratum_feature_edge_segment_template <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
inter_stratum_feature_edge_- segment_template to edge_segment_vertex (as start_vertex)	PATH			<pre> inter_stratum_feature_edge_segment_template <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
inter_stratum_feature_edge_- segment_template to trimmed_line (as reference_path)	PATH			<pre> inter_stratum_feature_edge_segment_template <= part_template_definition <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation.items[i] -> representation_item => geometric_representation_item => curve => bounded_curve => trimmed_curve </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTER_STRATUM_FEATURE_- EDGE_TEMPLATE	inter_stratum_feature_edge_- template	210		inter_stratum_feature_edge_template <= part_template_definition <= shape_aspect {shape_aspect shape_aspect.description = 'inter stratum feature edge template'}}
INTERCONNECT_MODULE	physical_unit	210	338, 268	physical_unit <= product_definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']}]}
interconnect_module to assembly_module (as assembly_design_requirement)	PATH			physical_unit <= product_definition <= product_definition.relationship.related_- product_definition {product_definition.relationship product_definition.relationship.name = 'assembly design requirement'} product_definition.relationship.related_- product_definition -> product_definition {product_definition [product_definition => physical_unit] [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']}]}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERCONNECT_MODULE_- EDGE	inter_stratum_feature	210	333	<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature} (shape_aspect (shape_aspect.description = 'plated interconnect module edge') (shape_aspect.description = 'interconnect module edge') (shape_aspect.description = 'partially plated interconnect module edge')) </pre>
INTERCONNECT_MODULE_- EDGE_SEGMENT	interconnect_module_edge_- segment	210		<pre> interconnect_module_edge_segment <= inter_stratum_feature </pre>
interconnect_module_edge_- segment to interconnect_module_edge (as composed_edge)	PATH		333	<pre> interconnect_module_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed edge'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect module edge'} shape_aspect => component_shape_aspect => inter_stratum_feature </pre>
interconnect_module_edge_- segment to edge_segment_vertex (as end_vertex)	PATH			<pre> interconnect_module_edge_segment <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
interconnect_module_edge_- segment to edge_segment_vertex (as start_vertex)	PATH			<pre> interconnect_module_edge_segment <= shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
interconnect_module_edge_- segment to inter_stratum_feature_edge_- segment_template (as usage_definition)	PATH			<pre> interconnect_module_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatiing_shape_- aspect -> {shape_aspect shape_aspect.description = 'inter stratum feature edge segment template'} shape_aspect => part_template_definition => inter_stratum_feature_edge_segment_template </pre>
INTERCONNECT_MODULE_- MACRO_COMPONENT	component_definition	210		<pre> component_definition <= product_definition {product_definition [product_definition.name = 'interconnect module']} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'macro occurrence'.]</pre>
interconnect_module_macro_- component to interconnect_module (as design_definition)	PATH			<pre>component_definition <= product_definition <- product_definition.relationship.related_- product_definition {product_definition.relationship product_definition.relationship.name = 'design definition'} product_definition.relationship product_definition.relationship.relatiing_- product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)</pre>
INTERCONNECT_MODULE_- MACRO_COMPONENT_JOIN_- TERMINAL	interconnect_module_macro_- component_join_terminal	210		<pre>interconnect_module_macro_component_join_- terminal <= component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'interconnect module component terminal'] [shape_aspect.product_definitional = .TRUE.]}</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_macro_- component_join_terminal to interconnect_module_macro_- component (as associated_component)	PATH			interconnect_module_macro_component_join_- terminal <= component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'macro occurrence']} product_definition => component_definition
interconnect_module_macro_- component_join_terminal to interconnect_module_join_- terminal (as definition)	PATH			interconnect_module_macro_component_join_- terminal <= component_terminal <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition product_definition.name = 'interconnect module'} shape_aspect => interconnect_module_terminal => interconnect_module_join_terminal </pre>
INTERCONNECT_MODULE_- STRATUM_ASSEMBLY_RELA- TIONSHIP	assembly_component_usage	44	192, 354, 339, 193	<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} </pre>
reference_designation	assembly_component - usage.reference_designator	44		
interconnect_module_stratum_- assembly_relationship to interconnect_module (as assembly)	PATH		263, 264, 268, 267, 265, 266, 338	<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatng_- product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'physical design'[]} product_definition => physical_unit
interconnect_module_stratum_- assembly_relationship to stratum (as component)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => stratum
INTERCONNECT_MODULE_- STRATUM_BASED_TERMINAL	interconnect_module_stratum_- based_terminal	210	333, 340, 268	interconnect_module_stratum_based_terminal <= interconnect_module_terminal <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'interconnect module'}
INTERFACE_ACCESS_MATERIAL_REMOVAL_LAMINATE_- COMPONENT	interface_access_material_- removal_component_shape_aspect	210		interface_access_material_removal_component_- shape_aspect <= material_removal_component_shape_aspect

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interface_access_material_- removal_laminate_component to laminate_component_interface_- terminal (as accessed_interface)	PATH			<pre> interface_access_material_removal_component_- shape_aspect <= material_removal_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'accessed interface'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'land interface terminal') (shape_aspect.description = 'non functional land interface terminal') (shape_aspect.description = 'component termination passage interface terminal') (shape_aspect.description = 'printed connector component interface terminal')} shape_aspect => laminate_component_interface_terminal </pre>
INTERFACE_ACCESS_STRATUM_- FEATURE_TEMPLATE_- COMPONENT	interface_access_component_- shape_aspect	210		<pre> interface_access_component_shape_aspect <= component_shape_aspect <= shape_aspect (shape_aspect.description = 'stratum feature template component') </pre>
interface_access_stratum_- feature_template_component to laminate_component_interface_- terminal (as accessed_interface)	PATH			<pre> interface_access_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.name = 'accessed interface'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect (shape_aspect.description = 'land interface terminal') (shape_aspect.description = 'non functional land interface terminal') (shape_aspect.description = 'component termination passage interface terminal') (shape_aspect.description = 'printed connector component interface terminal')} shape_aspect => lamineate_component_interface_terminal
INTERFACIAL_CONNECTION	plated_passage	210	333	plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'interfacial connection'}
INTERNAL_PROBE_ACCESS_- AREA	probe_access_area	210	340, 338, 268	probe_access_area <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'internal probe access area'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence']]}</pre>
internal_probe_access_area to stratum_feature (as stratum_feature_implementation)	PATH		333	<pre>probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'implementation'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_feature</pre>
INTERNAL_STRATUM_AC- CESS	shape_aspect_relationship	41	196, 197	<pre>{shape_aspect_relationship shape_aspect_relationship.name = 'internal stratum access'}</pre>
internal_stratum_access to internal_probe_access_area (as connected_probe)	PATH		340, 265, 266, 263, 264, 268, 267, 333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> {shape_aspect [shape_aspect.description = 'internal probe access area'] [shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -></pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component'[]]} shape_aspect => component_shape_aspect => probe_access_area </pre>
internal_stratum_access to via (as inter_stratum_feature)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via')} (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
JOIN_TWO_PHYSICAL_CONNECTIVITY_DEFINITION_SUPPORTING_INTER_STRATUM_FEATURE	plated_inter_stratum_feature	210		<pre> plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {[shape_aspect shape_aspect.description = 'join two physical connectivity definition supporting inter stratum feature']} [shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'join implementation'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.name = 'inter stratum join'} shape_aspect => join_shape_aspect}} </pre>
<pre> join_two_physical_connectivity_- definition_supporting_inter_- stratum_feature to physical_connectivity_definition (as joined_definitions) </pre>	PATH			<pre> plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition property_definition.relationship {property_definition_relationship property_definition_relationship.name = 'requirement implementation'} property_definition.relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => physical_connectivity_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
JOIN_TWO_PHYSICAL_CONNECTIVITY_SUPPORTING_PRINTED_COMPONENT	component_shape_aspect	210	338	<pre> component_shape_aspect <= shape_aspect { shape_aspect shape_aspect.description = 'join 2 physical connectivity definition supporting' } component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition property_definition.relationship { property_definition.relationship property_definition.relationship.name = 'requirement implementation' } property_definition.relationship.relatin_ property_definition -> property_definition property_definition.definition -> property_definition.definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => physical_connectivity_definition </pre>
join_two_physical_connectivity_definition_supporting_printed_component to physical_connectivity_definition (as joined_definitions)	PATH		338, 333	
LAMINATE_COMPONENT	(component_shape_aspect)	210	268, 333	<pre> component_shape_aspect <= shape_aspect { (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminate text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect (shape_aspect.description = 'laminate component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminate text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminate component')

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'stratum feature template component'))} {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition [product_definition product_definition.description = 'laminated component'] [product_definition => component_definition] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']}]}</pre>
laminated_component to external_definition (as cad_layer)	PATH			<pre>component_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'cad_layer'} property_definition <- property_definition_representation.definition property_definition_representation</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => externally_defined_representation_item => external_definition </pre>
laminare_component to part_template (as usage_definition)	PATH		338, 340, 333	<pre> component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => part_template_definition </pre>
LAMINATE_COMPONENT_- FEATURE	(component_terminal) (laminare_component_interface_- terminal)	210 210		<pre> (component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'via terminal') (shape_aspect.description = 'conductive interconnect element terminal') (shape_aspect.description = 'printed component join terminal')}} (laminare_component_interface_terminal <= shape_aspect {shape_aspect </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'land interface terminal') (shape_aspect.description = 'non functional land interface terminal') (shape_aspect.description = 'component termination passage interface terminal') (shape_aspect.description = 'printed connector component interface terminal'))
LAMINATE_COMPONENT_IN- TERFACE_TERMINAL	lamineate_component_interface_ terminal	210		lamineate_component_interface_terminal<= shape_aspect {shape_aspect (shape_aspect.description = 'land interface terminal') (shape_aspect.description = 'non functional land interface terminal') (shape_aspect.description = 'component termination passage interface terminal') (shape_aspect.description = 'printed connector component interface terminal')}}
LAMINATE_COMPONENT_- JOIN_TERMINAL	component_terminal	210		component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'embedded component terminal') (shape_aspect.description = 'printed component join terminal')}}
LAMINATE_GROUP_COMPO- NENT_MAKE_FROM_RELA- TIONSHIP	lamineate_group_component_ make_from_relationship	210		lamineate_group_component_make_from_ relationship <= make_from_usage_option

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
laminated_group_component_- make_from_relationship to assembly_group_component (as resultant_component)	PATH			laminated_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => assembly_group_component_definition
laminated_group_component_- make_from_relationship to assembly_group_component (as reusable_component)	PATH			laminated_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => assembly_group_component_definition
laminated_group_component_- make_from_relationship to stratum_make_from_relationship (as stratum_mapping)	PATH			laminated_group_component_make_from_- relationship <= make_from_usage_option <= product_definition_usage <= characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <-

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_relationship.relatiing_- property_definition {property_definition_relationship property_definition_relationship.name = 'stratum mapping'} property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => {product_definition_relationship product_definition_relationship.name = 'stratum make from'} product_definition_usage => make_from_usage_option </pre>
LAMINATE_TEXT_COMPO- NENT	component_shape_aspect	210	340, 338, 268	<pre> component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'laminare text component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']]}</pre>
of_character	shape_aspect.name	41		<pre>component_shape_aspect <= shape_aspect shape_aspect.name</pre>
laminated_text_component to part_text_template (as usage_definition)	PATH		333	<pre>shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => part_template_definition => part_text_template_definition</pre>
laminated_text_component to stratum_feature (as stratum_feature_implementation)	PATH		333	<pre>component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature_implementation'} shape_aspect_relationship.related_shape_- aspect -></pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => stratum_feature
LAMINATE_TEXT_STRING_- COMPONENT	array_placement_group_- component_shape_aspect	210		array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'laminated text string component'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']}]
message	shape_aspect.name	41		array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_aspect.name

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
of_horizontal_justification	descriptive_representation_item	45		<pre> array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'of horizontal justification'} representation_item => {(descriptive_representation_item.description = 'left') (descriptive_representation_item.description = 'right') (descriptive_representation_item.description = 'centre')}} descriptive_representation_item array_placement_group_component_shape_aspect <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition </pre>
of_vertical_justification	descriptive_representation_item	45		

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'of vertical justification'} representation_item => {(descriptive_representation_item.description = 'top') (descriptive_representation_item.description = 'centre') (descriptive_representation_item.description = 'bottom')} descriptive_representation_item </pre>
LAND_INTERFACE_TERMINAL	laminar_component_interface_ - terminal	210	333	<pre> laminar_component_interface_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'land interface terminal'} </pre>
land_interface_terminal to land (as associated_component)	PATH		333, 340, 268	<pre> laminar_component_interface_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminated component'] [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect (shape_aspect.description = 'contact size dependent land')} (shape_aspect.description = 'component termination passage and contact size dependent land') (shape_aspect.description = 'via and contact size dependent land')}} shape_aspect => component_shape_aspect => land </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
land_interface_terminal to land_template_interface_terminal (as definition)	PATH			<pre> lamineate_component_interface_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => land_template_terminal </pre>
LAND_JOIN_TERMINAL	component_terminal	210	333	<pre> component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'land join terminal'} </pre>
land_join_terminal to land (as associated_component)	PATH		333, 340, 268	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'lamineate component']} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition]] characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect (shape_aspect.description = 'functional land') (shape_aspect.description = 'via dependent land') (shape_aspect.description = 'component termination passage dependent land') (shape_aspect.description = 'contact size dependent land') (shape_aspect.description = 'component termination passage and contact size dependent land') (shape_aspect.description = 'via and contact size dependent land')}} shape_aspect => component_shape_aspect => land</pre>
land_join_terminal to land_template_join_terminal (as definition)	PATH			<pre>component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'instantiated feature' shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal' shape_aspect => land_template_terminal </pre>
LAYER	layer	210		<pre> layer <= shape_aspect </pre>
name	shape_aspect.name	41	333	<pre> layer <= shape_aspect shape_aspect.name </pre>
layer to external_definition (as layer_usage)	PATH			<pre> layer document_assigned_item = layer document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'layer definition'} </pre>
layer to inter_stratum_extent (as extent)	PATH		340	<pre> layer <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'}
LAYOUT_MACRO_COMPONENT	layout_macro_component	210		layout_macro_component <= printed_component <= component_definition <= product_definition
layout_macro_component to layout_macro_definition (as design_definition)	PATH			layout_macro_component <= printed_component <= component_definition <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'design definition'}
layout_macro_component to layout_macro_floor_plan_ template (as usage_definition)	PATH			product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => physical_unit => interconnect_definition => layout_macro_definition layout_macro_component <= printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_template_definition => layout_macro_floor_plan_template
LAYOUT_MACRO_DEFINI- TION	layout_macro_definition	210		layout_macro_definition <= interconnect_definition <= physical_unit <= product_definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']} layout_macro_definition <= interconnect_definition <= physical_unit <= product_definition characterized_product_definition = product_- definition characterized_product_definition <- property_definition.definition
layout_macro_definition to layout_macro_floor_plan_- template (as floor_layout_view)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'floor layout view'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_template_definition => layout_macro_floor_plan_template </pre>
LAYOUT_MACRO_DEFINITION_TERMINAL_TO_USAGE_TERMINAL_ASSIGNMENT	layout_macro_definition_terminal_to_usage_terminal_assignment	210		<pre> {[layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect] [layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship]} </pre>
layout_macro_definition_terminal_to_usage_terminal_assignment to printed_part_template_terminal (as defined_template_feature)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => printed_part_template_terminal </pre>
layout_macro_definition_terminal_to_usage_terminal_assignment to component_termination_passage_join_terminal (as assigned_design_object)	PATH			<pre> layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_macro_definition_- terminal_to_usage_terminal_- assignment to conductive_interconnect_- element_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to land_join_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to non_functional_land_join_- terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_macro_definition_- terminal_to_usage_terminal_- assignment to printed_component_join_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to via_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to embedded_component_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
LAYOUT_MACRO_FLOOR_- PLAN_TEMPLATE	layout_macro_floor_plan_- template	210		layout_macro_floor_plan_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'printed part template'}
layout_macro_floor_plan_- template to layout_macro_definition_- terminal_to_usage_terminal_- assignment (as access_maps)	PATH			layout_macro_floor_plan_template <= part_template_definition <= shape_aspect <= shape_aspect.relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'access maps'}
				shape_aspect <= shape_aspect_relationship.related_shape_- aspect -> shape_aspect => layout_macro_definition_terminal_to_usage_- terminal_assignment
MATERIAL_REMOVAL_LAMI- NATE_COMPONENT	material_removal_component_- shape_aspect	210		material_removal_component_shape_aspect <= component_shape_aspect
material_removal_laminate_- component to stratum_feature (as design_intent)	PATH			material_removal_component_shape_aspect <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'implementation'}
				shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_feature
material_removal_laminate_- component to material_removal_feature_- template (as usage_definition)	PATH			component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'instantiated template'}
				shape_aspect_relationship.related_shape_- aspect ->

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect {shape_aspect.description = 'material removal feature template'} shape_aspect => part_template_definition </pre>
MULTI_LAYER_MATERIAL_REMOVAL_LAMINATE_COMPONENT multi_layer_stratum_laminate_layer (as derived_layer_geometry)	multi_layer_material_removal_component_shape_aspect	210		<pre> multi_layer_material_removal_component_shape_- aspect <= material_removal_component_shape_aspect multi_layer_material_removal_component_shape_- aspect <= material_removal_component_shape_aspect <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived layer_geometry'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => layer </pre>
	PATH			
MULTI_LAYER_STRATUM_FEATURE multi_layer_stratum_feature to layer (as derived_layer_geometry)	multi_layer_stratum_feature	210		<pre> multi_layer_stratum_feature <= stratum_feature multi_layer_stratum_feature <= stratum_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived layer_geometry'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => layer </pre>
	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
MULTI_LAYER_STRATUM_FEATURE_TEMPLATE_COMPONENT	multi_layer_component_shape_aspect	210		multi_layer_component_shape_aspect <= component_shape_aspect <= shape_aspect (shape_aspect.description = 'stratum feature template component')
multi_layer_stratum_feature_template_component to layer (as derived_layer_geometry)	PATH			multi_layer_component_shape_aspect <= component_shape_aspect <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived layer geometry'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer
NEXT_HIGHER_ASSEMBLY_RELATIONSHIP	(next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage])	44 44 44	213, 214	(next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=] assembly_component_usage)
reference_designation	PATH			next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator
next_higher_assembly_relationship to data_element (as quantity)	PATH			next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage.quantity -> measure_with_unit

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
NON_CONDUCTIVE_BASE_- BLIND_VIA	plated_passage	210	333	plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'non conductive base blind via'}
NON_FUNCTIONAL_LAND_- INTERFACE_TERMINAL	lamineate_component_interface_- terminal	210	333	lamineate_component_interface_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'non functional land interface terminal'}
non_functional_land_interface_- terminal to land_template_interface_terminal (as definition)	PATH		333	lamineate_component_interface_terminal <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}
				shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interface terminal'}
				shape_aspect => land_template_terminal
non_functional_land_interface_- terminal to non_functional_land (as associated_component)	PATH		340, 268, 333	lamineate_component_interface_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition {[product_definition product_definition.description = 'laminated component'] [product_definition => component_definition]} characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'via and contact size dependent non functional land')} shape_aspect => component_shape_aspect => land
NON_FUNCTIONAL_LAND_ JOIN_TERMINAL	component_terminal	210		component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'non functional land join terminal'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
non_functional_land_join_- terminal to land_template_join_terminal (as definition)	PATH		333	<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => land_template_terminal </pre>
non_functional_land_join_- terminal to non_functional_land (as associated_component)	PATH		340, 268, 333	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminare component'] [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect (shape_aspect.description = 'non functional land') (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') (shape_aspect.description = 'via and contact size dependent non functional land')}} shape_aspect => component_shape_aspect => land
PADSTACK_OCCURRENCE	padstack_occurrence	210		padstack_occurrence <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect
padstack_occurrence to padstack_definition (as usage_definition)	PATH			padstack_occurrence <= assembly_group_component_shape_aspect <= component_shape_aspect <= shape_aspect <-

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'usage definition'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => padstack_definition
PADSTACK_OCCURRENCE_- SUB_ASSEMBLY_RELATION- SHIP	padstack_occurrence_shape_- aspect_relationship	210		padstack_occurrence_shape_aspect_relationship <= shape_aspect_relationship
sub_assembly_reference_designa- tion	PATH			padstack_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.name
padstack_occurrence_sub_- assembly_relationship to laminare_component (as related_component)	PATH			padstack_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
padstack_occurrence_sub_- assembly_relationship to padstack_occurrence (as relating_component)	PATH			padstack_occurrence_shape_aspect_relationship <= shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => assembly_group_component_shape_aspect <= component_shape_aspect => padstack_occurrence
PART_CONNECTED_TERMINALS_DEFINITION	part_connected_terminals_- definition	210		part_connected_terminals_definition <= shape_aspect
name	shape_aspect.name	41	333	part_connected_terminals_definition <= shape_aspect shape_aspect.name
part_connected_terminals_- definition to part_terminal (as connected_terminals)	PATH		333	part_connected_terminals_definition <= shape_aspect <= shape_aspect_relationship.relying_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'member connected_terminal'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal) (interconnect_module_terminal) (packaged_part_terminal)
part_connected_terminals_- definition to Assembly_module_usage_view (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => (physical_unit =>
part_connected_terminals_- definition to Bare_die (as associated_definition)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				bare_die (physical_unit => externally_defined_physical_unit => externally_defined_bare_die) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)
part_connected_terminals_- definition to Interconnect_module_usage_view (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design_usage']} product_definition => (physical_unit) (physical_unit externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PART_TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= { shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition product_definition.format -> product_definition.format product_definition.format.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'template definition']] </pre>
part_template_to_analytical_model (as part_template_analytical_model)	PATH		333, 340	<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model
part_template to analytical_representation (as analytical_part_template_ representation)	PATH			part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {[representation => analytical_model] [representation representation.name = 'part template analytical model']}] representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => analytical_representation

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_template to characteristic (as physical_characteristic)	PATH		333, 340, 109	<pre> part_template_definition <= {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type}
part_template to coordinated_characteristic (as physical_characteristic)	PATH			part_template_definition <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> (characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'})} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation ->

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation representation.items[i] -> representation_item => coordinated_representation_item part_template_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'} </pre>
part_template to template_definition (as associated_definition)	PATH			
PARTIALLY_PLATED_- CUTOUT	inter_stratum_feature	210		<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'partially plated cutout'} </pre>
PARTIALLY_PLATED_INTER- CONNECT_MODULE_EDGE	inter_stratum_feature	210		<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'partially plated interconnect module edge'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PCB	interconnect_definition	210	338, 268	<pre> interconnect_definition <= physical_unit <= product_definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] </pre>
pcb to pcb_usage_view (as usage_view)	PATH		338, 268	<pre> interconnect_definition <= physical_unit <= product_definition <- product_definition.relationship.related_ - product_definition product_definition.relationship {product_definition.relationship.name = 'design usage'} product_definition.relationship product_definition.relationship.relatin_ - product_definition -> product_definition {[product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']] product_definition => physical_unit => (interconnect_definition) (externally_defined_physical_unit => externally_defined_interconnect_definition) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(externally_defined_physical_unit => library_defined_physical_unit => library_defined_interconnect_definition)
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect.description = 'interconnect module component stratum based terminal'} (shape_aspect.description = 'minimally defined component terminal'}) }) (component_interface_terminal <= shape_aspect)
PHYSICAL_CONNECTIVITY_- INTERRUPTING_CUTOUT	inter_stratum_feature	210	333	inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'physical connectivity interrupting cutout'}
physical_connectivity_- interrupting_cutout to conductive_interconnect_element (as interrupted_connectivity_element)	PATH		333, 340, 338	inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'interrupted connectivity element'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect [(shape_aspect.description = 'conductive interconnect element with pre defined transitions')]

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition] [shape_aspect => component_shape_aspect]}</pre>
PHYSICAL_CONNECTIVITY_- MAKE_FROM_RELATIONSHIP	make_from_connectivity_- relationship	210		<pre>make_from_connectivity_relationship <= shape_aspect_relationship</pre>
physical_connectivity_make_- from_relationship to physical_connectivity_definition (as resultant_definition)	PATH		344, 333	<pre>make_from_connectivity_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition</pre>
physical_connectivity_make_- from_relationship to physical_connectivity_definition (as reusable_definition)	PATH			<pre>make_from_connectivity_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => physical_connectivity_definition</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PLATED_CONDUCTIVE_- BASE_BLIND_VIA	plated_passage	210	333	plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'plated conductive base blind via'}
PLATED_CUTOUT	plated_inter_stratum_feature	210		plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'plated cutout'}
PLATED_CUTOUT_EDGE_- SEGMENT	plated_cutout_edge_segment	210	333	[plated_cutout_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect] [plated_cutout_edge_segment <= cutout_edge_segment <= shape_aspect_relationship]
plated_cutout_edge_segment to partially_plated_cutout (as composed_cutout)	PATH			plated_cutout_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed cutout'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'partially plated cutout'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => component_shape_aspect => inter_stratum_feature
PLATED_INTER_STRATUM_FEATURE	plated_inter_stratum_feature	210		plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'plated cutout') (shape_aspect.description = 'plated cutout edge segment') (shape_aspect.description = 'plated interconnect module edge segment') interconnect module edge'}}}
PLATED_INTERCONNECT_MODULE_EDGE	plated_inter_stratum_feature	210	333	plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'plated interconnect module edge') interconnect module edge'}}}
PLATED_INTERCONNECT_MODULE_EDGE_SEGMENT	plated_interconnect_module_- edge_segment	210		[plated_interconnect_module_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect] [plated_interconnect_module_edge_segment <=

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				interconnect_module_edge_segment <= shape_aspect_relationship]
plated_interconnect_module_- edge_segment to partially_plated_interconnect_- module_edge (as composed_edge)	PATH			plated_interconnect_module_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed edge'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => {shape_aspect shape_aspect.description = 'partially plated interconnect module edge'} component_shape_aspect => inter_stratum_feature
PLATED_PASSAGE	plated_passage	210		plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PRIMARY_STRATUM_INDICATOR_SYMBOL	component_shape_aspect	210	333	component_shape_aspect <= shape_aspect {shape_aspect.description = 'primary stratum indicator symbol'}
PRINTED_COMPONENT	printed_component	210	338, 268	printed_component <= component_definition <= product_definition {product_definition [product_definition.description = 'printed component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']}]}
printed_component to external_definition (as cad_layer)	PATH			printed_component <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'cad_layer'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_component to layer (as required_material_stack)	PATH			externally_defined_representation_item => external_definition printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'required material_stack'} shape_aspect_relationship.related_shape_ aspect -> shape_aspect => layer
printed_component to printed_component_link (as printed_component_stack)	PATH			printed_component <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'printed_component_stack'}

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.related_ product_definition -> product_definition => printed_component_link
printed_component to printed_part_template (as usage_definition)	PATH		338, 340, 333	printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => part_template_definition
printed_component to stratum_feature (as stratum_feature_implementation)	PATH			printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature_implementation'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => stratum_feature
PRINTED_COMPONENT_ JOIN_TERMINAL	component_terminal	210		component_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'printed component join terminal'] [shape_aspect.product_definitional = .TRUE.]}
printed_component_join_terminal to printed_component (as associated_component)	PATH			component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition [product_definition.description = 'laminated component']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']] product_definition => component_definition => printed_component </pre>
printed_component_join_terminal to printed_part_template_join_ - terminal (as definition)	PATH		333	<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => printed_part_template_terminal </pre>
printed_component_join_terminal to stratum_feature (as stratum_concept_implementation)	PATH			<pre> component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'implementation' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature </pre>
PRINTED_COMPONENT_LINK	printed_component_link	210		<pre> printed_component_link <= [product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'printed component stack'} product_definition_relationship.relatng_- product_definition -> product_definition => component_definition => printed_component] [product_definition_relationship] </pre>
printed_component_link to printed_component (as precedent_printed_component)	PATH			<pre> printed_component_link <= product_definition_relationship product_definition_relationship.relatng_- product_definition -> product_definition => component_definition => printed_component </pre>
printed_component_link to printed_component (as subsequent_printed_component)	PATH			<pre> printed_component_link <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => component_definition => printed_component </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_component_link to printed_part_template_link (as definition)	PATH			<pre> printed_component_link <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'definition'} property_definition.relationship.relatin_ property_definition -> property_definition => stratum_technology_link => printed_part_template_link </pre>
PRINTED_CONNECTOR_COM- PONENT	printed_component	210	338, 268	<pre> printed_component <= component_definition <= product_definition {product_definition [product_definition.name = 'printed connector component'] [product_definition.description = 'printed component'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'layout occurrence']} </pre>
printed_connector_component to printed_connector_template (as usage_definition)	PATH		338, 340, 333	<pre> printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'printed connector template'} shape_aspect => part_template_definition </pre>
PRINTED_CONNECTOR_COMPONENT_INTERFACE_TERMINAL	lamineate_component_interface_terminal	210		<pre> lamineate_component_interface_terminal <= shape_aspect {shape_aspect [shape_aspect.description = 'printed connector component interface terminal'] [shape_aspect.product_definitional = .TRUE.]} </pre>
printed_connector_component_interface_terminal to printed_connector_component (as associated_component)	PATH		333, 340, 268, 338	<pre> lamineate_component_interface_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'printed connector component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element occurrence'[]} product_definition => component_definition => printed_component </pre>
printed_connector_component_- interface_terminal to printed_part_template_interface_- terminal (as definition)	PATH		333	<pre> laminatate_component_interface_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => printed_part_template_terminal laminatate_component_interface_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'implementation'} </pre>
printed_connector_component_- interface_terminal to stratum_feature (as stratum_concept_implementation)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature </pre>
PROBE_ACCESS_AREA	probe_access_area	210	333, 340, 338, 268	<pre> probe_access_area <= component_shape_aspect <= shape_aspect {[(shape_aspect.description = 'probe access area') (shape_aspect.description = 'internal probe access area')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition.description = 'lamineate component']]}} </pre>
probe_access_area to connection_zone (as probe_contact_zone)	PATH		333	<pre> probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'probe contact zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>
probe_access_area to stratum_feature (as probed_layout_item)	PATH			<pre> probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'probed layout item'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature </pre>
PROMISSORY_USAGE_IN_- PRODUCT_MODEL_RELATIONSHIP	applied_promissory_usage_in_- product_model_assignment	210		<pre> {applied_promissory_usage_in_product_model_- assignment <= group_assignment group_assignment.assigned_group -> group => promissory_usage_in_product_model} </pre>
promissory_usage_in_product_- model_relationship to Ee_product_definition (as component)	PATH			<pre> applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_definition (product_definition => product_definition_with_associated_documents) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (product_definition) (product_definition => evaluation_product_definition) (product_definition => physical_unit) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition => analytical_model_definition) (product_definition => footprint_definition) (product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => stratum) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit) applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_concept </pre>
promissory_usage_in_product_- model_relationship to ee_product_model (as product_model)	PATH			

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
promissory_usage_in_product_model_relationship to security_classification (as security_code)	PATH			<pre> applied_promissory_usage_in_product_model_assignment security_classification_assigned_item = applied_promissory_usage_in_product_model_assignment security_classification_assigned_item <- applied_security_classification_assignment.items[1] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_security_classification -> security_classification </pre>
PROMISSORY_USAGE_RELATIONSHIP	promissory_usage_occurrence	44	275, 276	
promissory_usage_relationship to physical_unit (as component)	PATH		354, 339, 338	<pre> promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design')} (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
promissory_usage_relationship to physical_unit_design_view (as assembly)	PATH			<pre> promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relati product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit </pre>
promissory_usage_relationship to security_classification (as security_code)	PATH			<pre> promissory_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification - assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification </pre>
ROUTED_PRINTED_COMPONENT	routed_printed_component	210	338	<pre> routed_printed_component <= printed_component <= component_definition <= product_definition {product_definition [product_definition.description = 'printed component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
routed_printed_component to curve (as routed_centrelines_shape)	PATH			<pre> routed_printed_component <= printed_component <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation.items[i] -> representation_item => geometric_representation_item => curve </pre>
routed_printed_component to inter_stratum_extent (as component_extent)	PATH		340, 338	<pre> routed_printed_component <= printed_component <= component_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {shape_aspect => component_shape_aspect} shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'component extent'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'} routed_printed_component <= printed_component <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition
routed_printed_component to printed_part_cross_section_ template (as usage_definition)	PATH		333	

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect => component_shape_aspect} shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'printed part cross section template'} shape_aspect => part_template_definition
SHIELD	(integral_shield)	210	338, 333	(integral_shield <= component_shape_aspect <= shape_aspect)
shield to assembly_component (as shielded_item)	PATH		338, 340, 333	(integral_shield <= component_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'shielded item'} product_definition_relationship product_definition_relationship.relatiing_ product_definition -> product_definition => component_definition (integral_shield <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'shielded item'} shape_aspect_relationship shape_aspect_relationship.relatiing_shape_ aspect -> shape_aspect => component_shape_aspect) (integral_shield <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'shielded item'} shape_aspect_relationship
shield to component_feature (as shielded_item)	PATH		338, 340, 333	shape_aspect_relationship

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_ aspect -> (shape_aspect => component_terminal) (shape_aspect => component_interface_terminal) (shape_aspect => laminat_component_interface_terminal) (shape_aspect => component_feature)) </pre>
shield to physical_connectivity_definition (as shielded_item)	PATH		338, 340, 333	<pre> (integral_shield <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'shielded item'}) shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => physical_connectivity_definition) </pre>
SPECIAL_SYMBOL_LAMI- NATE_COMPONENT	component_shape_aspect	210	340, 338, 268	<pre> component_shape_aspect <= shape_aspect {shape_aspect [(shape_aspect.description = 'special symbol laminat component') (shape_aspect.description = 'primary stratum indicator symbol')]} [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']] </pre>
special_symbol_laminated_- component to special_symbol_part_template (as usage_definition)	PATH		333, 340	<pre> component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect [shape_aspect.description = 'special symbol part template']} [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element[] shape_aspect => part_template_definition component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature implementation'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature </pre>
special_symbol_laminate_- component to stratum_feature (as stratum_feature_implementation)	PATH			
STRATUM	stratum	210	265, 266, 338	<pre> stratum <= product_definition {product_definition (product_definition.name = 'design layer') (product_definition.name = 'documentation layer') (product_definition.name = 'generic stratum')} </pre>
name	product_definition.id	41		<pre> stratum <= product_definition product_definition.id </pre>
stratum to external_definition (as stratum_usage)	PATH			<pre> (stratum document_assigned_item = stratum document_assigned_item <- applied_document_reference.items[i] </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'stratum definition'} (stratum <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition {property_definition.relationship property_definition_relationship.name = 'stratum usage'} property_definition.relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object characterized_object => external_definition </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum to stratum_technology (as of_technology)	PATH			<pre> stratum <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship property_definition_relationship.relatin_ property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => stratum_technology </pre>
STRATUM_AVERAGE_SUR- FACE	stratum_surface	210		<pre> {stratum_surface <= shape_aspect shape_aspect.description = 'average surface'} </pre>
stratum_average_surface to stratum (as of_stratum)	PATH			<pre> stratum_surface <= shape_aspect shape_aspect.of_shape -> product_definition_shape product_definition_shape <= </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition product_definition => stratum </pre>
STRATUM_FEATURE	stratum_feature	210	333, 322	stratum_feature <= shape_aspect
feature_of_size	descriptive_representation_item	45	342	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'feature of size'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true')} (descriptive_representation_item.description = 'false')} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
name	shape_aspect.name	41		stratum_feature <= shape_aspect shape_aspect.name
stratum_feature to stratum (as resident_stratum)	PATH		340, 265, 266, 338	stratum_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => stratum
STRATUM_FEATURE_IN - STRATUM_3D_POSITION	mapped_item	43	317, 337, 109	{mapped_item <= representation_item representation_item.name = 'positioned stratum feature shape in stratum'}
stratum_feature_in_stratum_3d - position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
stratum_feature_in_stratum_3d - position to stratum_3d_shape (as stratum_shape)	PATH		340, 265, 266, 338	mapped_item <= representation_item <= representation.items[i] {representation representation.name = '3d bound volume shape'} representation <= property_definition_representation.used_ representation

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => stratum[]} representation => shape_representation </pre>
<pre> stratum_feature_in_stratum_3d_- position to stratum_feature_non_planar_2d_- shape (as feature_shape) </pre>	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {representation representation.name = 'stratum feature non planar 2d shape'} representation => shape_representation => manifold_surface_shape_representation </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_feature_in_stratum_3d_position to stratum_feature_3d_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
STRATUM_FEATURE_IN- STRATUM_PLANAR_POSI- TION	mapped_item	43	109, 318	{mapped_item <= representation_item representation_item.name = 'positioned stratum feature shape in stratum'}
stratum_feature_in_stratum_ planar_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d
stratum_feature_in_stratum_ planar_position to stratum_feature_planar_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
stratum_feature_in_stratum_ planar_position to stratum_feature_styled_area_ representation (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_feature_in_stratum_- planar_position to stratum_planar_shape (as stratum_shape)	PATH		340, 265, 266, 338	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => stratum]]} representation => shape_representation </pre>
STRATUM_FEATURE_TEM- PLATE_COMPONENT	component_shape_aspect	210	333, 340, 338, 268, 290	<pre> component_shape_aspect <= shape_aspect {shape_aspect [(shape_aspect => component_shape_aspect => land) (shape_aspect.description = 'stratum feature template component')] [shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']] </pre>
stratum_feature_template_- component to stratum_feature (as stratum_feature_implementation)	PATH		333	<pre> component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'stratum feature_implementation'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_feature </pre>
stratum_feature_template_- component to stratum_feature_template (as usage_definition)	PATH			<pre> component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => {shape_aspect (shape_aspect.description = 'stratum feature template') (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based') (shape_aspect.description = 'default unsupported passage based') (shape_aspect.description = 'default component termination passage based')}} part_template_definition) (shape_aspect => part_template_definition => land_physical_template) </pre>
STRATUM_INTERCONNECT_- MODULE_3D_POSITION	mapped_item	43	337, 323, 109	<pre> {mapped_item <= representation_item representation_item.name = '3d positioned stratum shape in interconnect module'} </pre>
stratum_interconnect_module_- 3d_position to axis_placement_3d (as transformation)	PATH			<pre> mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_interconnect_module_- 3d_position to interconnect_module_stratum_- assembly_relationship (as stratum_component)	PATH		340, 339, 354	<pre> mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => placement => axis2_placement_3d} representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship {product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} product_definition_relationship => product_definition_usage => assembly_component_usage </pre>
stratum_interconnect_module_- 3d_position to physical_unit_3d_shape (as interconnect_module_shape)	PATH		340, 265, 266, 263, 264, 268, 267	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape'] representation <- property_definition_representation.used_- representation </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit}} representation => shape_representation </pre>
stratum_interconnect_module_- 3d_position to stratum_3d_shape (as stratum_shape)	PATH		340, 265, 266, 338	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin']} [representation_item => geometric_representation_item => axis2_placement_3d]} representation_map.mapped_representation -> {[representation representation.name = '3d bound volume shape']} [representation <- </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition definition product_definition => stratum[]} representation => shape_representation
STRATUM_INTERCONNECT_- MODULE_PLANAR_POSITION	mapped_item	43	109, 324	{mapped_item <= representation_item representation_item.name = 'planar positioned stratum shape in interconnect module'}
stratum_interconnect_module_- planar_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d
stratum_interconnect_module_- planar_position to interconnect_module_stratum_- assembly_relationship (as stratum_component)	PATH		340, 354	mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => placement => axis2_placement_2d} representation_item <-

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship {product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} product_definition_relationship => product_definition_usage => assembly_component_usage mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation </pre>
stratum_interconnect_module_- planar_position to physical_unit_planar_shape (as interconnect_module_shape)	PATH		340, 265, 266, 338, 263, 264, 268, 267	

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit[] representation => shape_representation </pre>
stratum_interconnect_module_- planar_position to stratum_planar_shape (as stratum_shape)	PATH		340, 265, 266, 338	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => axis2_placement_2d]} representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => stratum[]} representation => shape_representation </pre>
STRATUM_MAKE_FROM_RELATIONSHIP	make_from_usage_option	44	215, 325, 354, 339	<pre> {make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'stratum make from'} </pre>
stratum_make_from_relationship to stratum (as resultant_stratum)	PATH		265, 266, 338	<pre> make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatin_- product_definition -> product_definition => stratum </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_make_from_relationship to stratum (as reusable_stratum)	PATH			make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => stratum
STRATUM_SURFACE	stratum_surface	210	333, 329	stratum_surface <= shape_aspect
name	PATH			stratum_surface stratum_surface <= shape_aspect {shape_aspect (shape_aspect.description = 'primary surface') (shape_aspect.description = 'secondary surface')}
stratum_surface to stratum (as of_stratum)	PATH		340, 265, 266, 338	shape_aspect.description stratum_surface stratum_surface <= shape_aspect shape_aspect.of_shape -> product_definition_shape product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum
SUB_ASSEMBLY_RELATION- SHIP	(assembly_component_usage) (shape_aspect_relationship)	44 41		

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
sub_assembly_reference_designa- tion	(assembly_component_- usage.reference_designator) (shape_aspect_- relationship.description)	44 41		
sub_assembly_relationship to assembly_component (as related_component)	PATH			(assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship.related_- product_definition -> product_definition => component_definition) (shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect)
sub_assembly_relationship to assembly_group_component (as relating_component)	PATH		338	(assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'group component'} product_definition_relationship.related_- product_definition ->

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => component_definition => assembly_group_component_definition) (shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'group component'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => component_shape_aspect => assembly_group_component_shape_aspect) </pre>
THERMAL_COMPONENT	(thermal_component_shape_- aspect)	210	333, 340, 338	<pre> (thermal_component_shape_aspect <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => {product_definition product_definition.description = 'laminare component'} component_definition}) </pre>
THERMAL_ISOLATION_LAMINATE_COMPONENT	thermal_isolation_component_- shape_aspect	210		<pre> thermal_isolation_component_shape_aspect <= filled_area_material_removal_component_shape_- aspect </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
UNROUTED_CONDUCTIVE_- INTERCONNECT_ELEMENT	component_shape_aspect	210		component_shape_aspect <= shape_aspect (shape_aspect.description = 'unrouted conductive interconnect element')
unrouted_conductive_- interconnect_element to layer_connection_point_link (as connected_points)	PATH			component_shape_aspect <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'connected points'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => layer_connection_point_link
UNSUPPORTED_PASSAGE	inter_stratum_feature	210	333	inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect.description = 'unsupported passage'}
unsupported_passage to unsupported_passage_template (as usage_definition)	PATH		333	inter_stratum_feature <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
VIA	plated_passage	210	333	<pre> plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect (shape_aspect.description = 'bonded conductive base blind via')} (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')} } </pre>
via to via_template (as usage_definition)	PATH		333	<pre> plated_passage <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'via template'} shape_aspect => part_template_definition </pre>
VIA_TERMINAL	component_terminal	210	333	<pre> component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'via terminal'} </pre>

Table 60 – Mapping table for interconnect_module_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
via_terminal to via (as associated_component)	PATH		333	<pre> component_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {[product_definition product_definition.description = 'laminated component'] [product_definition => component_definition]} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition.definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection')}</pre>

Table 60 – Mapping table for interconnect_module_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage
via_terminal to via_template_terminal (as definition)	PATH			component_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'via template terminal'}

Table 61 – Mapping table for interconnect_module_edge UoF

Application element	AIM element	Source	Rules	Reference path
EDGE_SEGMENT_VERTEX	edge_segment_vertex	210	146, 147	edge_segment_vertex <= physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'single datum'}
name	descriptive_representation_- item.description	45		edge_segment_vertex <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.description = 'shape element characterization'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'shape element name'} representation.items[i] -> representative_item => descriptive_representation_item descriptive_representation_item.description
INTERCONNECT_MODULE_- EDGE	inter_stratum_feature	210	333	inter_stratum_feature <= component_shape_aspect <= shape_aspect

Table 61 – Mapping table for interconnect_module_edge UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature} (shape_aspect (shape_aspect.description = 'plated interconnect module edge') (shape_aspect.description = 'interconnect module edge') (shape_aspect.description = 'partially plated interconnect module edge')) </pre>
INTERCONNECT_MODULE_- EDGE_SEGMENT	interconnect_module_edge_- segment	210		interconnect_module_edge_segment <= inter_stratum_feature
interconnect_module_edge_- segment to interconnect_module_edge (as composed_edge)	PATH		333	<pre> interconnect_module_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed edge'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect module edge'} shape_aspect => component_shape_aspect => inter_stratum_feature </pre>
interconnect_module_edge_- segment to edge_segment_vertex (as end_vertex)	PATH			<pre> interconnect_module_edge_segment <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} </pre>

Table 61 – Mapping table for interconnect_module_edge UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
interconnect_module_edge_- segment to edge_segment_vertex (as start_vertex)	PATH			<pre> interconnect_module_edge_segment <= shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
interconnect_module_edge_- segment to inter_stratum_feature_edge_- segment_template (as usage_definition)	PATH			<pre> interconnect_module_edge_segment <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatiing_shape_- aspect -> {shape_aspect shape_aspect.description = 'inter stratum feature edge segment template'} shape_aspect => part_template_definition => inter_stratum_feature_edge_segment_template </pre>
PARTIALLY_PLATED_INTER- CONNECT_MODULE_EDGE	inter_stratum_feature	210		<pre> inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'partially plated interconnect module edge'} </pre>

Table 61 – Mapping table for interconnect_module_edge UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect.description = 'interconnect module component stratum based terminal'} (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PLATED_INTERCONNECT_- MODULE_EDGE	plated_inter_stratum_feature	210	333	plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'plated interconnect module edge'}
PLATED_INTERCONNECT_- MODULE_EDGE_SEGMENT	plated_interconnect_module_- edge_segment	210		[plated_interconnect_module_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect] [plated_interconnect_module_edge_segment <= interconnect_module_edge_segment <= shape_aspect_relationship]
plated_interconnect_module_- edge_segment to partially_plated_interconnect_- module_edge (as composed_edge)	PATH			plated_interconnect_module_edge_segment <= plated_inter_stratum_feature <= inter_stratum_feature <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship

Table 61 – Mapping table for interconnect_module_edge UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>shape_aspect_relationship.name = 'composed edge' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => {shape_aspect shape_aspect.description = 'partially plated interconnect module edge' component_shape_aspect => inter_stratum_feature</pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF

Application element	AIM element	Source	Rules	Reference path
EDGE_SEGMENT_VERTEX	edge_segment_vertex	210	146, 147	edge_segment_vertex <= physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'single datum'}
name	descriptive_representation_- item.description	45		edge_segment_vertex <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.description = 'shape element characterization'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'shape element name'} representation.items[i] -> representative_item => descriptive_representation_item descriptive_representation_item.description
INTERCONNECT_MODULE_- CAVITY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module cavity surface'}

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_cavity_- surface to interconnect_module_surface_- feature (as reference_surface)	PATH			<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference surface'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}} </pre>
INTERCONNECT_MODULE_- CUTOUT_SEGMENT_SUR- FACE	interconnect_module_cutout_- segment_surface	210		<pre> [interconnect_module_cutout_segment_surface <= shape_aspect] [interconnect_module_cutout_segment_surface <= shape_aspect_relationship] {shape_aspect.description = 'interconnect module cutout segment surface'} </pre>
interconnect_module_cutout_- segment_surface to interconnect_module_cutout_- surface (as composed_surface)	PATH			<pre> interconnect_module_cutout_segment_surface <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.relatng_shape_- aspect -> </pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_cutout_- segment_surface to edge_segment_vertex (as end_vertex)	PATH			{shape_aspect shape_aspect.description = 'interconnect module cutout surface'} shape_aspect interconnect_module_cutout_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
interconnect_module_cutout_- segment_surface to edge_segment_vertex (as start_vertex)	PATH			interconnect_module_cutout_segment_surface <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
INTERCONNECT_MODULE_- CUTOUT_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module cutout surface'}
INTERCONNECT_MODULE_- EDGE_SEGMENT_SURFACE	interconnect_module_edge_- segment_surface	210		[interconnect_module_edge_segment_surface <= shape_aspect] [interconnect_module_edge_segment_surface <= shape_aspect_relationship] {shape_aspect.description = 'interconnect module edge segment surface'}

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_edge_- segment_surface to interconnect_module_edge_- surface (as composed_surface)	PATH			interconnect_module_edge_segment_surface <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect module edge surface'} shape_aspect
interconnect_module_edge_- segment_surface to edge_segment_vertex (as end_vertex)	PATH			interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
interconnect_module_edge_- segment_surface to edge_segment_vertex (as start_vertex)	PATH			interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect =>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit_datum => edge_segment_vertex
INTERCONNECT_MODULE_- EDGE_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module edge surface'}
INTERCONNECT_MODULE_- INTERFACE_TERMINAL	interconnect_module_interface_- terminal	210		interconnect_module_interface_terminal <= interconnect_module_terminal <= shape_aspect
INTERCONNECT_MODULE_- JOIN_TERMINAL	interconnect_module_join_- terminal	210		interconnect_module_join_terminal <= interconnect_module_terminal <= shape_aspect
INTERCONNECT_MODULE_- PRIMARY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module primary surface'}
INTERCONNECT_MODULE_- SECONDARY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module secondary surface'}
INTERCONNECT_MODULE_- STRATUM_BASED_TERMINAL	interconnect_module_stratum_- based_terminal	210	333, 340, 268	interconnect_module_stratum_based_terminal <= interconnect_module_terminal <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition product_definition characterized_product_definition definition product_definition.name = 'interconnect module'}

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERCONNECT_MODULE_- SURFACE_FEATURE	shape_aspect	41		<pre> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}} </pre>
interconnect_module_surface_- feature to interconnect_module_usage_view (as associated_definition)	PATH		194	<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']}] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) </pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(physical_unit => externally_defined_physical_unit => library_defined_physical_unit)
INTERCONNECT_MODULE_TERMINAL	interconnect_module_terminal	210	333, 340, 268	interconnect_module_terminal <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition product_definition.name = 'interconnect module'}
interconnect_module_terminal to connection_zone (as terminal_connection_zone)	PATH		333	interconnect_module_terminal <= shape_aspect <= shape_aspect.relationship.relatng_shape_ aspect shape_aspect.relationship {shape_aspect.relationship shape_aspect.relationship.name = 'terminal connection zone'} shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
interconnect_module_terminal to interconnect_module_usage_view (as associated_definition)	PATH		333, 340, 268, 338	interconnect_module_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)
INTERCONNECT_MODULE_ TERMINAL_SURFACE_ CONSTITUENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'interconnect module terminal surface constituent relationship'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => interconnect_module_terminal
interconnect_module_terminal_ surface_constituent_relationship to interconnect_module_terminal (as related_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => interconnect_module_terminal
interconnect_module_terminal_ surface_constituent_relationship to interconnect_module_surface_ feature (as relating_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface')}

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}</pre>
INTERCONNECT_MODULE_- USAGE_VIEW	<pre>(physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)</pre>	<pre>210 210 210</pre>	268	<pre>(physical_unit <= product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}</pre>
interconnect_module_usage_view to functional_unit_usage_view (as implemented_function)	PATH		338	<pre>(physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=)</pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'implemented function'} product_definition_relationship.relatiing_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional design usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition </pre>
interconnect_module_usage_view to located_interconnect_module_- thickness_requirement (as located_thickness_requirement)	PATH			

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition [property_definition.name = 'located interconnect module thickness']} property_definition <= property_definition_representation.definition property_definition_representation (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition
interconnect_module_usage_view to length_data_element (as minimum_thickness_over_ metal_requirement)	PATH			

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition {property_definition property_definition.name = 'interconnect module usage view physical characteristics'} property_definition <= property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum thickness over metal requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition
interconnect_module_usage_view to length_data_element (as maximum_thickness_over_ metal_requirement)	PATH			

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition {property_definition property_definition.name = 'interconnect module usage view physical characteristics'} property_definition <= property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum thickness over metal requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition
interconnect_module_usage_view to length_data_element (as minimum_thickness_over_ dielectric_requirement)	PATH			

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition {property_definition property_definition.name = 'interconnect module usage view physical characteristics'} property_definition <= property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum thickness over dielectric requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition
interconnect_module_usage_view to length_data_element (as maximum_thickness_over_ dielectric_requirement)	PATH			

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition {property_definition property_definition.name = 'interconnect module usage view physical characteristics'} property_definition <= property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum thickness over dielectric requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition
interconnect_module_usage_view to restraint_condition (as measurement_condition)	PATH			(physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition property_definition {property_definition (property_definition.description = 'restraint')} (property_definition.description = 'tolerance specific restraint') </pre>
LOCATED_INTERCONNECT_- MODULE_THICKNESS_RE- QUIREMENT	property_definition_representation	41	205	<pre> property_definition_representation {property_definition_representation.definition -> property_definition [property_definition.name = 'located interconnect module thickness']} [property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= </pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'physical design usage'[]}] </pre>
located_interconnect_module_- thickness_requirement to axis_placement (as reference_location)	PATH			<pre> property_definition_representation property_definition_representation.used_- representation -> representation <- representation_relationship.rep_2 {representation_relationship representation_relationship.name = 'reference location'} representation_relationship.rep_1 -> representation representation.items[i] -> (representation_item => geometric_representation_item => placement => axis2_placement_2d) (representation_item => geometric_representation_item => placement => axis2_placement_3d) </pre>
located_interconnect_module_- thickness_requirement to length_data_element (as maximum_thickness_requirement)	PATH			<pre> property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum thickness'} representation_item => </pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
located_interconnect_module_- thickness_requirement to length_data_element (as minimum_thickness_requirement)	PATH			<pre> property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
PCB_USAGE_VIEW	<pre> (interconnect_definition) (externally_defined_interconnect_- definition) (library_defined_interconnect_- definition) </pre>	210 210 210	268	<pre> (interconnect_definition <= physical_unit <= product_definition) (externally_defined_interconnect_definition <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_interconnect_definition <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> </pre>

Table 62 – Mapping table for interconnect_module_usage_view UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'physical design usage'[]}]}

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_REQUIREMENT_- ALLOCATION	(product_definition_relationship) (shape_aspect_relationship)	41 41		{product_definition_relationship product_definition_relationship.name = 'assembly_requirement_allocation'} {shape_aspect_relationship shape_aspect_relationship.name = 'assembly requirement_allocation'}
assembly_requirement_allocation to assembly_component (as assembly_assignment)	PATH		17, 338, 340, 333	product_definition_relationship product_definition_relationship.related_- product_definition -> (product_definition => component_definition) (product_definition {product_definition => component_definition} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect)
assembly_requirement_allocation to assembly_joint (as assembly_assignment)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_requirement_allocation to component_feature (as assembly_assignment)	PATH		17, 333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature)
assembly_requirement_allocation to ee_requirement_occurrence (as requirement)	PATH		17	(product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'design requirement') (application_context_element.name = 'requirement')} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => requirements_property) (shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect.of_shape -> product_definition_shape <= property_definition => requirements_property)

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_SEPARATION_REQUIREMENT_ALLOCATION	(product_definition_relationship) (shape_aspect_relationship)	41 41		{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'assembly_separation_requirement']}) {shape_aspect_relationship [shape_aspect_relationship.name = 'assembly requirement_allocation'] [shape_aspect_relationship.description = 'assembly_separation_requirement']})
assembly_separation_requirement_allocation to assembly_component (as disjoint_assignment)	PATH		17, 338, 340, 333	product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition => component_definition) ([product_definition => component_definition] [product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect])
assembly_separation_requirement_allocation to assembly_joint (as disjoint_assignment)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_separation_- requirement_allocation to component_feature (as disjoint_assignment)	PATH		17, 18, 333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminar_component_interface_terminal) (component_feature)
ELECTROMAGNETIC COM- PATIBILITY REQUIREMENT_- ALLOCATION	product_definition_relationship	41		{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'electromagnetic compatibility requirement allocation']}
LAYOUT LAND WIDTH TOL- ERANCE REQUIREMENT OC- CURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'layout land width tolerance requirement'}
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_negative_deviation)	PATH			requirements_property <= property_definition <= property_definition_representation.definition property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum negative deviation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
layout_land_width_tolerance_- requirement_occurrence to pitch_class (as mating_feature_pitch_class)	PATH			requirements_property <= property_definition <= property_definition_representation.definition property_definition_representation.used_- representation -> representation {representation representation.name = 'pitch class'}

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
LAYOUT_LINE_WIDTH_TOLERANCE_REQUIREMENT_OCCURRENCE	requirements_property	210		<pre> requirements_property <= property_definition {property_definition property_definition.description = 'layout line width tolerance requirement'} requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum negative deviation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_negative_deviation)	PATH			
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_positive_deviation)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum negative deviation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
layout_line_width_tolerance_- requirement_occurrence to line_width_class (as tolerance_line_width_class)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation.items[i] -> representation_item => styled_item </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_line_width_tolerance_- requirement_occurrence to stratum_technology (as characterized_stratum_- technology)	PATH			requirements_property <= property_definition <- property_definition_relationship.related_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object characterized_object => stratum_technology
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
REQUIREMENT_ALLOCATION	(product_definition_relationship) (shape_aspect_relationship) (requirement_allocation_group)	41 41 210		{product_definition_relationship product_definition_relationship.name = 'requirement_allocation'} {shape_aspect_relationship shape_aspect_relationship.name = 'requirement allocation'} {[requirement_allocation_group <=

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_relationship[[requirement_allocation_group <= group]] requirement_allocation_group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = product) (group_assigned_item = product_definition_ formation) (group_assigned_item = configuration_item)) </pre>
requirement_allocation to stratum_surface (as target_product_object)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_surface </pre>
requirement_allocation to component_external_reference (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = representation {representation.name = 'component external reference'} </pre>
requirement_allocation to component_feature_external_- reference (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment group_assigned_item = representation_item {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item </pre>
requirement_allocation to component_feature (as target_product_object)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (component_terminal) (component_interface_terminal) (laminate_component_interface_terminal) (component_feature) </pre>
requirement_allocation to ee_product (as target_product_object)	PATH		268	<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product </pre>
requirement_allocation to ee_product_configuration (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = configuration_item </pre>
requirement_allocation to ee_product_definition (as target_product_object)	PATH			<pre> product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => (product_definition_with_associated_documents) (functional_unit) </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
requirement_allocation to ee_product_version (as target_product_object)	PATH			requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product_definition_formation
requirement_allocation to ee_requirement_occurrence (as allocated_requirement)	PATH			(product_definition_relationship product_definition_relationship.relati- product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'}) characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition product_definition <- characterized_definition <- property_definition.definition property_definition => (shape_aspect_relationship shape_aspect_relationship.relati- shape_aspect ->

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => (requirement_allocation_group <= property_definition_relationship property_definition_relationship.relatin_ g_ property_definition -> property_definition => requirements_property
requirement_allocation to ee_requirement_occurrence (as target_product_object)	PATH			requirement_allocation_group <= property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'derived from operation'} property_definition_relationship.related_ property_definition ->

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
requirement_allocation to assembly_component (as target_product_object)	PATH		333	<pre> property_definition => requirements_property product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition => component_definition) ([product_definition => component_definition] [product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect]) </pre>
requirement_allocation to inter_stratum_feature (as target_product_object)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature </pre>
requirement_allocation to laminar_component (as target_product_object)	PATH		340, 333, 268, 338	<pre> product_definition_relationship product_definition_relationship.related_ product_definition -> (product_definition {[product_definition </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition.description = 'laminated component' [product_definition => component_definition => printed_component]] (product_definition {[product_definition product_definition.description = 'laminated component'] [product_definition => component_definition]] characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {(shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminated text string component'}) => </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect (shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component'))} </pre>
requirement_allocation to physical_connectivity_definition (as target_product_object)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_definition </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
requirement_allocation to physical_connectivity_element (as target_product_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => physical_connectivity_element
requirement_allocation to stratum (as target_product_object)	PATH		338	product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => stratum
requirement_allocation to stratum_feature (as target_product_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature
SHIELD_ALLOCATION	product_definition_relationship	41	17, 309	{product_definition_relationship [product_definition_relationship.name = 'assembly requirement allocation'] [product_definition_relationship.description = 'shield allocation']}
shield_allocation to electromagnetic_requirement_ occurrence (as requirement)	PATH		17	product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship product_definition_relationship.name = 'shield group member'} product_definition characterized_product_definition = product_ definition characterized_product_definition = characterized_ product_definition characterized_definition <-

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition {property_definition property_definition.name = 'electromagnetic requirement'} property_definition => requirements_property </pre>
shield_allocation to shield (as assembly_assignment)	PATH		340, 333	<pre> product_definition.relationship product_definition.relationship.related_- product_definition -> product_definition (product_definition {product_definition => component_definition} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => component_shape_aspect => integral_shield) </pre>
SPECIFICATION_ALLOCATION	applied_document_reference	210		<pre> applied_document_reference <= document_reference </pre>
specification_allocation to assembly_component (as target_product_object)	PATH		333, 340, 338	<pre> applied_document_reference applied_document_reference.items[i] -> document_assigned_item (document_assigned_item = component_- definition) (document_assigned_item = component_shape_- aspect) </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
specification_allocation to component_feature (as target_product_object)	PATH		333	applied_document_reference applied_document_reference.items[i] -> document_assigned_item (document_assigned_item = component_terminal) (document_assigned_item = laminate_component_- interface_terminal) (document_assigned_item = component_- interface_terminal) (document_assigned_item = component_feature)
specification_allocation to ee_product (as target_product_object)	PATH			applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = product product
specification_allocation to ee_product_configuration (as target_product_object)	PATH			applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = configuration_item configuration_item
specification_allocation to ee_product_definition (as target_product_object)	PATH		338	applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = product_definition product_definition => (product_definition_with_associated_documents) (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit)
specification_allocation to ee_product_version (as target_product_object)	PATH			applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = product_definition_- formation product_definition_formation

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
specification_allocation to ee_specification (as allocated_specification)	PATH			applied_document_reference <= document_reference document_reference.assigned_document -> document => ee_specification
specification_allocation to inter_stratum_feature (as target_product_object)	PATH		333, 340, 338, 268	applied_document_reference document_assigned_item document_assigned_item = inter_stratum_feature inter_stratum_feature {inter_stratum_feature <= component_shape_aspect <= shape_aspect [(shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'plated cutout') (shape_aspect.description = 'plated cutout edge segment') (shape_aspect.description = 'plated interconnect module edge segment') (shape_aspect.description = 'unsupported passage') (shape_aspect.description = 'cutout') (shape_aspect.description = 'physical connectivity interrupting cutout') (shape_aspect.description = 'dielectric material passage')

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'cutout edge segment') (shape_aspect.description = 'interconnect module edge segment') (shape_aspect.description = 'interconnect module edge') [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition product_definition.description = 'laminare component'] [product_definition => component_definition]]</pre>
specification_allocation to laminare_component (as target_product_object)	PATH		338, 268, 333	<pre>applied_document_reference applied_document_reference.items[i] -> document_assigned_item (document_assigned_item = product_definition {[product_definition product_definition.description = 'laminare component'] [product_definition => component_definition => printed_component]}) (document_assigned_item = product_definition {[product_definition product_definition.description = 'laminare component']</pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition => component_definition]] characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {(shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) shape_aspect {shape_aspect.description = 'laminare text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect (shape_aspect.description = 'laminare component')</pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminare text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminare component') (shape_aspect.description = 'stratum feature template component'))))
specification_allocation to physical_connectivity_definition (as target_product_object)	PATH			applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = physical_- connectivity_definition physical_connectivity_definition
specification_allocation to physical_connectivity_element (as target_product_object)	PATH			applied_document_reference applied_document_reference.items[i] -> document_assigned_item document_assigned_item = physical_- connectivity_element physical_connectivity_element

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
specification_allocation to stratum (as target_product_object)	PATH			<pre> applied_document_reference document_assigned_item document_assigned_item = stratum stratum {stratum <= product_definition} </pre>
specification_allocation to stratum_feature (as target_product_object)	PATH			<pre> applied_document_reference document_assigned_item document_assigned_item = stratum_feature stratum_feature {stratum_feature <= shape_aspect} </pre>
THERMAL_ISOLATION RE- QUIREMENT_ASSIGNMENT	property_definition	41		<pre> property_definition property_definition.name = 'thermal isolation requirement assignment' </pre>
thermal_isolation_requirement_- assignment to ee_requirement_occurrence (as angular_orientation_requirement)	PATH			<pre> property_definition <- property_definition.relationship.related_- property_definition {property_definition.relationship property_definition.relationship.name = 'angular orientation requirement'} property_definition.relationship.related_- property_definition -> property_definition => requirements_property </pre>
thermal_isolation_requirement_- assignment to ee_requirement_occurrence (as effective_current_capacity_- requirement)	PATH			<pre> property_definition <- property_definition.relationship.related_- property_definition {property_definition.relationship property_definition.relationship.name = 'effective current capacity requirement'} property_definition.relationship.related_- property_definition -> property_definition => requirements_property </pre>

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
thermal_isolation_requirement_- assignment to ee_requirement_occurrence (as thermal_isolation_spacing_- requirement)	PATH			property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'spacing_requirement'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property
thermal_isolation_requirement_- assignment to thermal_isolation_removal_- structure (as resulting_structure)	PATH			property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'thermal isolation removal structure'} shape_aspect => shape_modification
thermal_isolation_requirement_- assignment to thermal_isolation_removal_- template (as resulting_structure)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'material removal feature template'} shape_aspect => part_template_definition => thermal_isolation_removal_template_definition
THERMAL_REQUIREMENT_- ALLOCATION	product_definition_relationship	41	360	{product_definition_relationship [product_definition_relationship.name = 'assembly_requirement_allocation'] [product_definition_relationship.description = 'thermal_requirement_allocation']}]}

Table 63 – Mapping table for interconnect_physical_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
thermal_requirement_allocation to thermal_requirement_occurrence (as requirement)	PATH		17	product_definition_relationship product_definition_relationship.relatiing_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'design requirement'} product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition {property_definition property_definition.name = 'thermal requirement'} property_definition => requirements_property

Table 64 – Mapping table for interconnect_placement_requirements UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_EXTERNAL_- REFERENCE	representation	43	33	{representation representation.name = 'component external reference'}
design_owner	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'design owner'} representation_item => descriptive_representation_item
part_number	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'part number'} representation_item => descriptive_representation_item
revision_code	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'revision code'} representation_item => descriptive_representation_item
product_definition_id	descriptive_representation_item	45		representation representation.items[i] -> {representation_item representation_item.name = 'product definition id'} representation_item => descriptive_representation_item
reference_designation	descriptive_representation_item	45		representation representation.items[i] ->

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item representation_item.name = 'reference designation'} representation_item => descriptive_representation_item
COMPONENT_FEATURE_EX- TERNAL_REFERENCE	descriptive_representation_item	45	34, 35	descriptive_representation_item <= {representation_item representation_item.name = 'component feature external reference'} representation_item
name	descriptive_representation_- item.description	45		
component_feature_external_- reference to component_external_reference (as associated_component)	PATH			descriptive_representation_item <= representation_item <- representation.items[i] {representation representation.name = 'component external reference'} representation
COMPONENT_GROUP_AS- SIGNMENT	(product_definition_relationship) (shape_aspect_relationship)	41 41		{product_definition_relationship product_definition_relationship.name = 'group member'} {shape_aspect_relationship shape_aspect_relationship.name = 'group member'}
component_group_assignment to assembly_component (as assigned_component)	PATH		265, 266, 338, 333	(product_definition_relationship product_definition_relationship.relation_- product_definition -> product_definition => component_definition) (shape_aspect_relationship)

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.relying_shape_aspect -> shape_aspect => component_shape_aspect)
component_group_assignment to placement_group (as associated_group)	PATH		265, 266, 338, 333	(product_definition_relationship product_definition -> product_definition => component_definition => group_product_definition) (shape_aspect_relationship shape_aspect_relationship.relying_shape_aspect -> shape_aspect => group_shape_aspect)
COMPONENT_PLACEMENT_ RESTRICTION_ASSIGNMENT	representation	43		representation {representation.name = 'component placement restriction assignment'}
components_permitted	descriptive_representation_item	45	109	representation {representation.items[i] -> {representation_item representation_item.name = 'components permitted'}} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}}
component_placement_ restriction_assignment to datum_based_length_measure (as maximum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation representation {representation.name = 'maximum mounting clearance'}

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_negative_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'maximum negative component height'}
component_placement_- restriction_assignment to datum_based_length_measure (as maximum_positive_- component_height)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'maximum positive component height'}
component_placement_- restriction_assignment to datum_based_length_measure (as minimum_mounting_clearance)	PATH			representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'minimum mounting clearance'}
component_placement_- restriction_assignment to item_restricted_requirement_- occurrence (as requirement)	PATH		42, 340	representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition => requirements_property => grouped_requirements_property {[grouped_requirements_property <= group group.name = 'item restricted requirements property']}

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_placement_- restriction_assignment to mounting_restriction_area (as area)	PATH		42, 340, 333	<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement_restriction_assignment area'} representation_relationship.rep_2 -> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_area </pre>
component_placement_- restriction_assignment to mounting_restriction_volume (as volume)	PATH			<pre> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'component placement_restriction_assignment volume'} representation_relationship.rep_2 -> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => mounting_restriction_volume
INTERCONNECT_MODULE_- COMPONENT_SURFACE_FEA- TURE	interconnect_module_- component_surface_feature	210	333	interconnect_module_component_surface_feature <= shape_aspect {shape_aspect.description = 'interconnect module component surface feature'} interconnect_module_component_surface_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}}
interconnect_module_- component_surface_feature to interconnect_module_surface_- feature (as definition)	PATH			

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERCONNECT_MODULE_- CONSTRAINT_REGION	group_shape_aspect	210	333	group_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'interconnect module constraint region'}
constrained_design_object_cat- egory	interconnect_module_design_- object_category	210	333, 340	group_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition.relationship.relationship_ property_definition {property_definition.relationship property_definition.relationship.name = 'constrained object'} property_definition.relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object {characterized_object [(characterized_object.description = 'cutout category') (characterized_object.description = 'fill area category') (characterized_object.description = 'inter stratum feature category') (characterized_object.description = 'stratum feature category')]

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.description = 'via category')[[characterized_object => interconnect_module_design_object_category]]
design_specific_purpose	descriptive_representation_item	45	109	group_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'design specific purpose'} representation_item => descriptive_representation_item descriptive_representation_item.description
keepout	descriptive_representation_item	45	333, 109	group_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation ->

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation representation.items[i] -> {representation_item representation_item.name = 'keepout'} representation_item => descriptive_representation_item {descriptive_representation_item.description = 'true'} (descriptive_representation_item.description = 'false')}</pre>
interconnect_module_constraint_- region to component_termination_passage (as non_conformant_interconnect_- module_design_object)	PATH		333	<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage</pre>
interconnect_module_constraint_- region to conductor (as non_conformant_interconnect_- module_design_object)	PATH		333	<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.relatng_shape_- aspect -></pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect shape_aspect.description = 'conductor'} shape_aspect => stratum_feature </pre>
interconnect_module_constraint_- region to conductive_filled_area (as non_conformant_interconnect_- module_design_object)	PATH			<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductive filled area'} shape_aspect => stratum_feature </pre>
interconnect_module_constraint_- region to cutout (as non_conformant_interconnect_- module_design_object)	PATH		333	<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature </pre>
interconnect_module_constraint_- region to ee_requirement_occurrence (as requirement)	PATH		340	<pre> group_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'requirement'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property </pre>
interconnect_module_constraint_- region to inter_stratum_extent (as associated_stratum_extent)	PATH		340, 265, 266, 338	<pre> group_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'associated_stratum_extent'} </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.relationship.relatiing_- property_definition -> property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition.relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'} </pre>
interconnect_module_constraint_- region to inter_stratum_feature (as non_conformant_interconnect_- module_design_object)	PATH		333	<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatiing_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature </pre>
interconnect_module_constraint_- region to interconnect_module (as scope)	PATH		340, 265, 266, 338, 263, 264, 268, 267	<pre> group_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']} product_definition => physical_unit </pre>
interconnect_module_constraint_- region to laminar_component (as non_conformant_interconnect_- module_design_object)	PATH		333, 340, 263, 264, 268, 267, 265, 266	<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect.description = 'laminated text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect { (shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component')}} (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => {product_definition.description = 'laminated component'} component_definition)
interconnect_module_constraint_ region to land (as non_conformant_interconnect_ module_design_object)	PATH		265, 266, 338, 263, 264, 268, 267, 333	group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect [(shape_aspect.description = 'non functional land') (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') (shape_aspect.description = 'via and contact size dependent non functional land')]}

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.name = 'interconnect module']]]} shape_aspect => component_shape_aspect => land</pre>
interconnect_module_constraint_- region to non_functional_land (as non_conformant_interconnect_- module_design_object)	PATH		333, 340, 265, 266, 338, 263, 264, 268, 267	<pre>group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect [(shape_aspect.description = 'non functional land')]</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') (shape_aspect.description = 'via and contact size dependent non functional land') [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.name = 'interconnect module']] shape_aspect => component_shape_aspect => land </pre>
interconnect_module_constraint_- region to stratum_feature (as non_conformant_interconnect_- module_design_object)	PATH			<pre> group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_constraint_- region to via (as non_conformant_interconnect_- module_design_object)	PATH		333	<pre> shape_aspect => stratum_feature group_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraint region violation'}} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via')} (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')}} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
ITEM_RESTRICTED_RE- QUIREMENT_OCCURRENCE	grouped_requirements_property	210		<pre> grouped_requirements_property <= [group {group group.name = 'item restricted requirements property'}}] </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
item_restricted_requirement_- occurrence to component_external_reference (as basis)	PATH			[requirements_property <= property_definition] grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = representation {representation representation.name = 'component external reference'} representation
item_restricted_requirement_- occurrence to component_feature_external_- reference (as basis)	PATH			grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = representation_item {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item
item_restricted_requirement_- occurrence to ee_requirement_occurrence (as basis)	PATH			grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = requirements_property requirements_property
item_restricted_requirement_- occurrence to assembly_component (as basis)	PATH		333, 340, 265, 266, 338	grouped_requirements_property <= group <= group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = component_definition) (group_assigned_item = component_shape_aspect) grouped_requirements_property <= group <= group_assignment.assigned_group group_assignment =>
item_restricted_requirement_- occurrence to component_feature (as basis)	PATH		333	applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = component_terminal) (group_assigned_item = component_interface_- terminal) (group_assigned_item = laminate_component_- interface_terminal) (group_assigned_item = component_feature) grouped_requirements_property <= group <= group_assignment.assigned_group group_assignment =>
item_restricted_requirement_- occurrence to inter_stratum_feature (as basis)	PATH		333, 340, 265, 266, 263, 264, 268, 267	grouped_requirements_property <= group <= group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = inter_stratum_feature

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> inter_stratum_feature { inter_stratum_feature <= component_shape_aspect <= shape_aspect [(shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'buried via') (shape_aspect.description = 'component termination passage') (shape_aspect.description = 'plated cutout') (shape_aspect.description = 'plated cutout edge segment') (shape_aspect.description = 'plated interconnect module edge segment') (shape_aspect.description = 'unsupported passage') (shape_aspect.description = 'cutout') (shape_aspect.description = 'physical connectivity interrupting cutout') (shape_aspect.description = 'dielectric material passage') (shape_aspect.description = 'cutout edge segment') (shape_aspect.description = 'interconnect module edge segment') (shape_aspect.description = 'interconnect module edge')] [shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition product_definition.description = 'laminare component'] [product_definition => component_definition]]} </pre>
item_restricted_requirement_- occurrence to laminare_component (as basis)	PATH		265, 266, 338, 263, 264, 268, 267, 333	<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = component_definition {component_definition <= product_definition product_definition.description = 'laminare component'}) (group_assigned_item = component_shape_aspect (component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (material_removal_component_shape_aspect)) (component_shape_aspect <= shape_aspect {shape_aspect.description = 'laminare text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (component_shape_aspect <= shape_aspect {(shape_aspect.description = 'laminare component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminare text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminare component') (shape_aspect.description = 'stratum feature template component')})) </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
item_restricted_requirement_- occurrence to part_feature (as basis)	PATH			<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = shape_aspect shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}} </pre>
item_restricted_requirement_- occurrence to stratum (as basis)	PATH			<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = stratum stratum {stratum <= product_definition} </pre>
item_restricted_requirement_- occurrence to stratum_feature (as basis)	PATH			<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = stratum_feature {stratum_feature <= shape_aspect} </pre>
item_restricted_requirement_- occurrence to stratum_surface (as basis)	PATH			<pre> grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = stratum_surface stratum_surface </pre>
LAYER_QUALIFIED_LAYOUT_SPACING_REQUIREMENT_OCCURRENCE	grouped_requirements_property	210		<pre> grouped_requirements_property <= [group {group group.name = 'layer qualified layout spacing requirements property'}}] [requirements_property <= property_definition] </pre>
layer_qualified_layout_spacing_- requirement to inter_stratum_extent (as layer_context)	PATH			<pre> grouped_requirements_property <= property_definition <- property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'qualified extent'} property_definition_relationship.related_- property_definition -> property_definition </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'} product_definition_relationship </pre>
LAYOUT_SPACING_REQUIRE- MENT_NON_CONFORMING_- DESIGN_OBJECT	shape_aspect_relationship	41	202	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'layout spacing requirement violation'} shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition => requirements_property => grouped_requirements_property {grouped_requirements_property <= group group.name = 'layout spacing requirements property'} </pre>
layout_spacing_requirement_- non_conforming_design_object to layout_spacing_requirement_- occurrence (as design_requirement)	PATH			
layout_spacing_requirement_- non_conforming_design_object to assembly_component (as non_conformant_design_object)	PATH		333, 340, 265, 266, 338	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => component_shape_aspect) (shape_aspect {shape_aspect (component_terminal) (component_interface_terminal) </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (laminate_component_interface_terminal) (component_feature) (interconnect_module_component_surface_- feature)}} shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition) </pre>
layout_spacing_requirement_- non_conforming_design_object to assembly_component (as reference_design_object)	PATH		333, 340, 265, 266, 338	<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => component_shape_aspect) (shape_aspect {shape_aspect => (component_terminal) (component_interface_terminal) (laminate_component_interface_terminal) (component_feature)}} shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition)
layout_spacing_requirement_ non_conforming_design_object to component_feature (as non_conformant_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_feature))
layout_spacing_requirement_ non_conforming_design_object to component_feature (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminated_component_interface_terminal) (component_shape_aspect))
layout_spacing_requirement_ non_conforming_design_object to component_termination_passage (as non_conformant_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_spacing_requirement_- non_conforming_design_object to component_termination_passage (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage
layout_spacing_requirement_- non_conforming_design_object to conductive_filled_area (as non_conformant_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductive filled area'} shape_aspect => stratum_feature
layout_spacing_requirement_- non_conforming_design_object to conductive_filled_area (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductive filled area'} shape_aspect => stratum_feature
layout_spacing_requirement_- non_conforming_design_object to conductor (as non_conformant_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'conductor'} shape_aspect => stratum_feature

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_spacing_requirement_- non_conforming_design_object to conductor (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> {shape_aspect shape_aspect.description = 'conductor'} shape_aspect => stratum_feature
layout_spacing_requirement_- non_conforming_design_object to cutout (as non_conformant_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature
layout_spacing_requirement_- non_conforming_design_object to cutout (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> {shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature
layout_spacing_requirement_- non_conforming_design_object to inter_stratum_feature (as non_conformant_design_object)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature
layout_spacing_requirement_- non_conforming_design_object to inter_stratum_feature (as reference_design_object)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect shape_aspect.description = 'cutout'} shape_aspect => component_shape_aspect => inter_stratum_feature

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_spacing_requirement_- non_conforming_design_object to laminated_component (as non_conformant_design_object)	PATH		333, 340, 263, 264, 268, 267, 265, 266, 338	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminated text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect {(shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component')}) (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => {product_definition.description = 'laminated component'}) component_definition)</pre>
layout_spacing_requirement_- non_conforming_design_object to laminated_component (as reference_design_object)	PATH		333, 340, 263, 264, 268, 267, 265, 266, 338	<pre>shape_aspect_relationship shape_aspect_relationship.relativing_shape_- aspect -> (shape_aspect => component_shape_aspect => (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area)</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminated text string component'}} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect {(shape_aspect.description = 'laminated component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminated text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') (shape_aspect.description = 'special symbol laminated component') (shape_aspect.description = 'stratum feature template component')}} </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => {product_definition.description = 'laminated component'} component_definition)</pre>
layout_spacing_requirement_- non_conforming_design_object to land (as non_conformant_design_object)	PATH		265, 266, 338, 263, 264, 268, 267, 333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect [(shape_aspect.description = 'functional land') (shape_aspect.description = 'via dependent land') (shape_aspect.description = 'component termination passage dependent land') (shape_aspect.description = 'contact size dependent land') (shape_aspect.description = 'component termination passage and contact size dependent land')]</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'via and contact size dependent land')]] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] shape_aspect => component_shape_aspect => land</pre>
layout_spacing_requirement_- non_conforming_design_object to land (as reference_design_object)	PATH		333, 340, 265, 266, 338, 263, 264, 268, 267	<pre>shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect [(shape_aspect.description = 'functional land') (shape_aspect.description = 'via dependent land') (shape_aspect.description = 'component termination passage dependent land') (shape_aspect.description = 'contact size dependent land') (shape_aspect.description = 'component termination passage and contact size dependent land') (shape_aspect.description = 'via and contact size dependent land')]]</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] shape_aspect => component_shape_aspect => land</pre>
layout_spacing_requirement_- non_conforming_design_object to non_functional_land (as non_conformant_design_object)	PATH		340, 265, 266, 338, 263, 264, 268, 267, 333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect [(shape_aspect.description = 'non functional land') (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') }</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'via and contact size dependent non functional land') [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] shape_aspect => component_shape_aspect => land</pre>
layout_spacing_requirement_- non_conforming_design_object to non_functional_land (as reference_design_object)	PATH		340, 265, 266, 338, 263, 264, 268, 267, 333	<pre>shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect [(shape_aspect.description = 'non functional land') (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'component termination passage and contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') }</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'via and contact size dependent non functional land') [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] shape_aspect => component_shape_aspect => land</pre>
layout_spacing_requirement_- non_conforming_design_object to stratum_feature (as non_conformant_design_object)	PATH			<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature</pre>
layout_spacing_requirement_- non_conforming_design_object to stratum_feature (as reference_design_object)	PATH		333	<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_feature</pre>
layout_spacing_requirement_- non_conforming_design_object to via (as non_conformant_design_object)	PATH			<pre>shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')}</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
layout_spacing_requirement_- non_conforming_design_object to via (as reference_design_object)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')}} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
LAYOUT_SPACING_REQUIRE- MENT_OCCURRENCE	grouped_requirements_property	210	203	<pre> grouped_requirements_property <= [requirements_property] [{group group.name = 'layout spacing requirements property'}]} </pre>
of_spacing_type	PATH			<pre> grouped_requirements_property <= group {(group.description = 'nearest boundary') (group.description = 'centroid') (group.description = 'furthest boundary')}} group.description </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
dependent_design_object_category	characterized_object.name	210	340	<pre> grouped_requirements.property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment group_assigned_item group_assigned_item = design_object design_object <= characterized_object {characterized_object.description = 'dependent design object category'} characterized_object.name {(characterized_object.name = 'assembly module assembly component') (characterized_object.name = 'assembly module macro component join terminal') (characterized_object.name = 'assembly module macro component') (characterized_object.name = 'assembly module') (characterized_object.name = 'bonded conductive base blind via') (characterized_object.name = 'buried via') (characterized_object.name = 'cable component') (characterized_object.name = 'component feature') (characterized_object.name = 'component mounting feature') (characterized_object.name = 'component termination passage and contact size dependent land') </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'component termination passage and contact size dependent non functional land') (characterized_object.name = 'component termination passage dependent land') (characterized_object.name = 'component termination passage dependent non functional land') (characterized_object.name = 'component termination passage interface terminal') (characterized_object.name = 'component termination passage join terminal') (characterized_object.name = 'component termination passage') (characterized_object.name = 'conductive interconnect element terminal') (characterized_object.name = 'conductive interconnect element with pre defined transitions') (characterized_object.name = 'conductive interconnect element with user defined single transition') (characterized_object.name = 'connected area component') (characterized_object.name = 'contact size dependent land') (characterized_object.name = 'contact size dependent non functional land') (characterized_object.name = 'cutout edge segment') (characterized_object.name = 'cutout') (characterized_object.name = 'dielectric material passage') (characterized_object.name = 'electrical isolation laminate component') (characterized_object.name = 'embedded component terminal')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'fiducial') (characterized_object.name = 'fill area') (characterized_object.name = 'filled area material removal laminate component') (characterized_object.name = 'inter stratum feature') (characterized_object.name = 'interconnect component interface terminal') (characterized_object.name = 'interconnect module assembly component') (characterized_object.name = 'interconnect module component stratum based terminal') (characterized_object.name = 'interconnect module component surface feature') (characterized_object.name = 'interconnect module edge segment') (characterized_object.name = 'interconnect module edge') (characterized_object.name = 'interconnect module macro component join terminal') (characterized_object.name = 'interconnect module macro component') (characterized_object.name = 'interface access material removal laminate component') (characterized_object.name = 'interface access stratum feature template component') (characterized_object.name = 'interface component') (characterized_object.name = 'interfacial connection') (characterized_object.name = 'internal probe access area') (characterized_object.name = 'join two physical connectivity definition supporting inter stratum feature')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'join two physical connectivity definition supporting printed component') (characterized_object.name = 'laminated component') (characterized_object.name = 'laminated text component') (characterized_object.name = 'laminated text string component') (characterized_object.name = 'land interface terminal') (characterized_object.name = 'land join terminal') (characterized_object.name = 'land') (characterized_object.name = 'material removal laminate component') (characterized_object.name = 'minimally defined component terminal') (characterized_object.name = 'movable packaged component join terminal') (characterized_object.name = 'multi layer material removal laminate component') (characterized_object.name = 'multi layer stratum feature template component') (characterized_object.name = 'non conductive base blind via') (characterized_object.name = 'non functional land interface terminal') (characterized_object.name = 'non functional land join terminal') (characterized_object.name = 'non functional land') (characterized_object.name = 'packaged connector component interface terminal') (characterized_object.name = 'packaged connector component')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'partially plated cutout') (characterized_object.name = 'partially plated interconnect module edge') (characterized_object.name = 'physical component') (characterized_object.name = 'physical connectivity interrupting cutout') (characterized_object.name = 'physical laminar component') (characterized_object.name = 'physical network supporting inter stratum feature') (characterized_object.name = 'plated conductive base blind via') (characterized_object.name = 'plated cutout edge segment') (characterized_object.name = 'plated cutout') (characterized_object.name = 'plated interconnect module edge segment') (characterized_object.name = 'plated interconnect module edge') (characterized_object.name = 'plated passage or unsupported passage') (characterized_object.name = 'plated passage') (characterized_object.name = 'primary stratum indicator symbol') (characterized_object.name = 'printed component join terminal') (characterized_object.name = 'printed connector component interface terminal') (characterized_object.name = 'probe access area') (characterized_object.name = 'routed interconnect component') (characterized_object.name = 'routed physical component')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (characterized_object.name = 'special symbol laminare component') (characterized_object.name = 'stratum feature template component') (characterized_object.name = 'stratum feature') (characterized_object.name = 'thermal isolation laminare component') (characterized_object.name = 'unrouted conductive interconnect element') (characterized_object.name = 'unsupported passage dependent non functional land') (characterized_object.name = 'unsupported passage') (characterized_object.name = 'via and contact size dependent land') (characterized_object.name = 'via and contact size dependent non functional land') (characterized_object.name = 'via dependent land') (characterized_object.name = 'via dependent non functional land') (characterized_object.name = 'via terminal') (characterized_object.name = 'via') } grouped_requirements_property <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = design_object design_object <= characterized_object {characterized_object characterized_object.description = 'reference design object category'} </pre>
reference_design_object_category	characterized_object.name	210		

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_object.name {(characterized_object.name = 'assembly module assembly component') (characterized_object.name = 'assembly module macro component join terminal') (characterized_object.name = 'assembly module macro component') (characterized_object.name = 'assembly module') (characterized_object.name = 'bonded conductive base blind via') (characterized_object.name = 'buried via') (characterized_object.name = 'cable component') (characterized_object.name = 'component feature') (characterized_object.name = 'component mounting feature') (characterized_object.name = 'component termination passage and contact size dependent land') (characterized_object.name = 'component termination passage and contact size dependent non functional land') (characterized_object.name = 'component termination passage dependent land') (characterized_object.name = 'component termination passage dependent non functional land') (characterized_object.name = 'component termination passage interface terminal') (characterized_object.name = 'component termination passage join terminal') (characterized_object.name = 'component termination passage')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'conductive interconnect element terminal') (characterized_object.name = 'conductive interconnect element with pre defined transitions') (characterized_object.name = 'conductive interconnect element with user defined single transition') (characterized_object.name = 'connected area component') (characterized_object.name = 'contact size dependent land') (characterized_object.name = 'contact size dependent non functional land') (characterized_object.name = 'cutout edge segment') (characterized_object.name = 'cutout') (characterized_object.name = 'dielectric material passage') (characterized_object.name = 'electrical isolation laminate component') (characterized_object.name = 'embedded component terminal') (characterized_object.name = 'fiducial') (characterized_object.name = 'fill area') (characterized_object.name = 'filled area material removal laminate component') (characterized_object.name = 'inter stratum feature') (characterized_object.name = 'interconnect component interface terminal') (characterized_object.name = 'interconnect module assembly component') (characterized_object.name = 'interconnect module component stratum based terminal') (characterized_object.name = 'interconnect module component surface feature')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'interconnect module edge segment') (characterized_object.name = 'interconnect module edge') (characterized_object.name = 'interconnect module macro component join terminal') (characterized_object.name = 'interconnect module macro component') (characterized_object.name = 'interface access material removal laminate component') (characterized_object.name = 'interface access stratum feature template component') (characterized_object.name = 'interface component') (characterized_object.name = 'interfacial connection') (characterized_object.name = 'internal probe access area') (characterized_object.name = 'join two physical connectivity definition supporting inter stratum feature') (characterized_object.name = 'join two physical connectivity definition supporting printed component') (characterized_object.name = 'laminate component') (characterized_object.name = 'laminate text component') (characterized_object.name = 'laminate text string component') (characterized_object.name = 'land interface terminal') (characterized_object.name = 'land join terminal') (characterized_object.name = 'land') (characterized_object.name = 'material removal laminate component')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'minimally defined component terminal') (characterized_object.name = 'movable packaged component join terminal') (characterized_object.name = 'multi layer material removal laminate component') (characterized_object.name = 'multi layer stratum feature template component') (characterized_object.name = 'non conductive base blind via') (characterized_object.name = 'non functional land interface terminal') (characterized_object.name = 'non functional land join terminal') (characterized_object.name = 'non functional land') (characterized_object.name = 'packaged connector component interface terminal') (characterized_object.name = 'packaged connector component') (characterized_object.name = 'partially plated cutout') (characterized_object.name = 'partially plated interconnect module edge') (characterized_object.name = 'physical component') (characterized_object.name = 'physical connectivity interrupting cutout') (characterized_object.name = 'physical laminate component') (characterized_object.name = 'physical network supporting inter stratum feature') (characterized_object.name = 'plated conductive base blind via') (characterized_object.name = 'plated cutout edge segment')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_object.name = 'plated cutout') (characterized_object.name = 'plated interconnect module edge segment') (characterized_object.name = 'plated interconnect module edge') (characterized_object.name = 'plated passage or unsupported passage') (characterized_object.name = 'plated passage') (characterized_object.name = 'primary stratum indicator symbol') (characterized_object.name = 'printed component join terminal') (characterized_object.name = 'printed connector component interface terminal') (characterized_object.name = 'probe access area') (characterized_object.name = 'routed interconnect component') (characterized_object.name = 'routed physical component') (characterized_object.name = 'special symbol laminate component') (characterized_object.name = 'stratum feature template component') (characterized_object.name = 'stratum feature') (characterized_object.name = 'thermal isolation laminate component') (characterized_object.name = 'unrouted conductive interconnect element') (characterized_object.name = 'unsupported passage dependent non functional land') (characterized_object.name = 'unsupported passage') (characterized_object.name = 'via and contact size dependent land')

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(characterized_object.name = 'via and contact size dependent non functional land') (characterized_object.name = 'via dependent land') (characterized_object.name = 'via dependent non functional land') (characterized_object.name = 'via terminal') (characterized_object.name = 'via')}</pre>
layout_spacing_requirement_- occurrence to tolerance_characteristic (as required_coordinated_- characteristic)	PATH			<pre>requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item {representation_item (representation_item.name = 'tolerance') (representation_item.name = 'minimum tolerance') (representation_item.name = 'maximum tolerance') (representation_item.name = 'plus minus tolerance') (representation_item.name = 'symmetrical tolerance') (representation_item.name = 'statistical tolerance')}</pre>
MOUNTING_RESTRICTION_- AREA	mounting_restriction_area	210	333	<pre>mounting_restriction_area <= shape_aspect mounting_restriction_area <= shape_aspect <- shape_aspect_relationship.relation_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mounting surface'}</pre>
mounting_restriction_area to interconnect_module_- component_surface_feature (as mounting_surface)	PATH			<pre>shape_aspect_relationship.related_shape_aspect -></pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.description = 'interconnect module component surface feature'} shape_aspect => interconnect_module_component_surface_feature
mounting_restriction_area to assembly_module (as scope)	PATH			mounting_restriction_area <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']} product_definition => physical_unit
MOUNTING_RESTRICTION_- VOLUME	mounting_restriction_volume	210		mounting_restriction_volume <= shape_aspect
mounting_restriction_volume to interconnect_module_- component_surface_feature (as mounting_surface)	PATH			mounting_restriction_volume <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mounting surface'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect.description = 'interconnect module component surface feature'} shape_aspect => interconnect_module_component_surface_feature mounting_restriction_volume <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'assembly module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] product_definition => physical_unit</pre>
mounting_restriction_volume to assembly_module (as scope)	PATH			
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		<pre>(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)</pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PLACEMENT_GROUP	(group_product_definition) (group_shape_aspect)	210 210	265, 266, 338, 333, 340	(group_product_definition <= component_definition <= product_definition {product_definition [product_definition.description = 'placement group'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'design requirement']}) (group_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'placement group'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => component_definition => group_product_definition]})
design_specific_usage	product.name	41	267, 268, 263, 264	(group_product_definition <= component_definition <= (group_shape_aspect <= shape_aspect

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition) product_definition product_definition.format -> product_definition.format product_definition.format.of_product -> product product.name
placement_group to ee_requirement_occurrence (as requirement)	PATH		265, 266, 338, 333, 340	(group_product_definition <= component_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition) (group_shape_aspect <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition) characterized_definition <= property_definition.definition {property_definition property_definition.name = 'requirement'} property_definition <-

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'requirement'} property_definition.relationship.relatin_ property_definition -> property_definition => requirements_property </pre>
PLACEMENT_GROUP_AREA_- ASSIGNMENT	shape_aspect_relationship	41	245	<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'area impacted group'} </pre>
placement_group_area_- assignment to mounting_restriction_area (as area)	PATH		333	<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => mounting_restriction_area </pre>
placement_group_area_- assignment to placement_group (as placed_group)	PATH		340, 265, 266, 338	<pre> shape_aspect_relationship.relatin_shape_- aspect -> (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element requirement'}) </pre>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => component_definition => group_product_definition) (shape_aspect => group_shape_aspect)
PLACEMENT_GROUP_VOLUME_ASSIGNMENT	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'volume impacted group'}
placement_group_volume_assignment to mounting_restriction_area (as volume)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => mounting_restriction_volume
placement_group_volume_assignment to placement_group (as placed_group)	PATH			shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'design requirement'} product_definition => component_definition =>

Table 64 – Mapping table for interconnect_placement_requirements UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				group_product_definition (shape_aspect => group_shape_aspect)

Table 65 – Mapping table for interconnect_shape UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_COMPONENT_- 2D_SHAPE	shape_representation	41	111, 340, 338, 333	<pre>{shape_representation <= [representation representation.name = 'planar projected shape'] [representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect)]}</pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_component_2d_shape to assembly_component (as shape_characterized_component)	PATH		340, 338, 333	<pre> shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect) </pre>
ASSEMBLY_COMPONENT_- 3D_SHAPE	(shape_representation) (manifold_surface_shape_- representation)	41 210	111, 340, 338, 333	<pre> (shape_representation <= representation {representation representation.name = '3d bound volume shape'}) </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (manifold_surface_shape_representation <= shape_representation <= representation {representation representation.name = 'open shell based surface'} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect) </pre>
assembly_component_3d_shape to assembly_component (as shape_characterized_component)	PATH		340, 338, 333	<pre> (shape_representation <= representation <-) (manifold_surface_shape_representation <= shape_representation <= representation <-) property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect)
ASSEMBLY_COMPONENT_- NON_PLANAR_SHAPE	manifold_subsurface_shape_ representation	521		manifold_subsurface_shape_representation <= shape_representation <= representation {representation representation.name = 'manifold subsurface'} representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_component_non_- planat_shape to assembly_component (as shape_characterized_component)	PATH			<pre> manifest_subsurface_shape_representation <= shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition {component_definition.description = 'laminated component'}) (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect) </pre>
BOND_ASSEMBLY_POSITION	mapped_item	43		<pre> {mapped_item <= representation_item representation_item.name = 'assembly position'} </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
bond_assembly_position to assembled_with_bonding (as bond)	PATH		109, 344	mapped_item <= representation_item <- representation.items[i] {representation representation.name = 'bond assembly position'} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'assembled with bonding'} shape_aspect_relationship => assembly_joint
bond_assembly_position to axis_placement_3d (as bond_definition_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
bond_assembly_position to default_assembly_bond_shape (as bond_model)	PATH		339, 354, 340	mapped_item mapped_item.mapping_source -> representation_map representation_map.mapping_representation -> representation <- property_definition_representation.used_ representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition [property_definition.description = 'default assembly bond shape']} [property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage[]}] property_definition => product_definition_shape mapped_item <= representation_item <- representation.items[i] {representation [representation.name = '3d bound volume shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition </pre>
bond_assembly_position to physical_unit_3d_shape (as assembly_model)	PATH		109, 340, 338, 111	

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation
COMPONENT_2D_EDGE_LO- CATION	component_location	210	342, 343	component_location <= representation {representation [representation.context_of_items -> representation.context => geometric_representation.context geometric_representation.context.coordinate_ space_dimension = 2] [representation.context representation.context.context_type = 'component edge']} }
component_2d_edge_location to assembly_joint (as mounting_surface_assembly_ joint)	PATH			component_location <= representation representation <= property_definition_representation.used_ representation property_definition_representation {property_definition_representation property_definition_representation.name = 'mounting_surface_assembly_joint'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_definition = shape_aspect shape_aspect => assembly_joint </pre>
component_2d_edge_location to assembly_joint (as reference_terminal_assembly_- joint)	PATH			<pre> component_location <= representation representation <- property_definition_representation.used_ - representation property_definition_representation {property_definition_representation property_definition_representation.name = 'reference_terminal_assembly_joint'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => assembly_joint </pre>
COMPONENT_2D_EMBED- DED_LOCATION	component_location	210		<pre> component_location <= representation {representation.context_of_items -> representation.context => [representation.context geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 2] representation_context representation_context.context_type = 'component_embedded'} </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
component_2d_embedded_- location to adjacent_stratum_surface_- embedded_component_surface_- definition (as precedent_surface)	PATH			<pre> component_location <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'precedent surface'}} representation_relationship.rep_2 -> representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'adjacent stratum_surface_embedded_component_surface'}} </pre>
component_2d_embedded_- location to adjacent_stratum_surface_- embedded_component_surface_- definition (as subsequent_surface)	PATH			<pre> component_location <- representation_relationship.rep_1 representation_relationship {representation_relationship representation_relationship.name = 'adjacent surface'}} representation_relationship.rep_2 -> representation <- property_definition_representation.used_- representation </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'adjacent stratum surface embedded component surface'} </pre>
component_2d_embedded_- location to fabrication_joint (as reference_terminal)	PATH			<pre> component_location <= representation representation <- property_definition_representation.used_- representation property_definition_representation {property_definition_representation property_definition_representation.name = 'reference terminal fabrication joint'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect shape_aspect => fabrication_joint </pre>
COMPONENT_2D_LOCATION	component_location	210		<pre> component_location <= representation ({representation representation.context_of_items -> representation.context representation_context => </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_context geometric_representation_context.coordinate_ space_dimension = 2}}
placement_fixed	descriptive_representation_ item.description	45		component_location <= representation representation.items[i] -> {representation_item representation_item.name = 'placement fixed'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}}}
component_2d_location to cartesian_transformation_ operator_2d (as transformation)	PATH		342	component_location <= representation representation.items[i] -> representation_item => geometric_representation_item => cartesian_transformation_operator => cartesian_transformation_operator_2d
COMPONENT_3D_LOCATION	component_location	210	342, 343	component_location <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_ space_dimension = 3}
placement_fixed	descriptive_representation_item	45	342	component_location <= representation representation.items[i] -> {representation_item representation_item.name = 'placement fixed'}

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = ,true') (descriptive_representation_item.description = ,false')} </pre>
component_3d_location to axis_placement_3d (as transformation)	PATH			<pre> component_location <= representation representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>
COMPONENT_ASSEMBLY - 2D_POSITION	mapped_item	43	308	<pre> {mapped_item <= representation_item representation_item.name = 'component assembly 2d position'} </pre>
component_assembly_2d_position to assembly_component_2d_shape (as component_model)	PATH		340, 338, 333, 111	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => axis2_placement_2d]} representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape']} [representation representation.items[i] -> representation_item => </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_representation_item => axis2_placement_2d] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect[]}] representation => shape_representation mapped_item <= {mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => </pre>
component_assembly_2d_position to component_2d_location (as transformation)	PATH		109, 342	

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				cartesian_transformation_operator => cartesian_transformation_operator_2d} representation_item <- representation.items[i] {representation representation.context_of_items -> representation.context => geometric_representation.context geometric_representation.context.coordinate_ space_dimension = 2} representation => component_location} representation_item <- representation.items[i] {representation representation.context_of_items -> representation.context => geometric_representation.context.coordinate_ space_dimension = 2} representation => component_location
component_assembly_2d_position to next_higher_assembly_ relationship (as component)	PATH		340, 339, 354	mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition ->

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage])
component_assembly_2d_position to physical_unit_planar_shape (as assembly_model)	PATH		338, 111	mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit]] representation => shape_representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COMPONENT_ASSEMBLY_- 3D_POSITION	mapped_item	43	308	{mapped_item <= representation_item representation_item.name = 'component assembly 3d position'}
component_assembly_3d_position to assembly_component_3d_shape (as component_model)	PATH		340, 338, 333, 111	mapped_item mapped_item.mapping_source -> representation_map {representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => placement => axis2_placement_3d]} representation_map.mapped_representation -> {[representation (representation.name = '3d bound volume shape') (representation.name = 'part template non planar 2d shape')] [representation representation.items[i] -> representation_item representation_item.name = 'origin'] [representation representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [representation <- property_definition_representation.used_- representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect[]}] representation => shape_representation mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => </pre>
component_assembly_3d_position to component_3d_location (as transformation)	PATH		342	

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> placement => axis2_placement_3d{ representation_item <- representation.items[i] representation => component_location </pre>
component_assembly_3d_position to next_higher_assembly_- relationship (as component)	PATH		340, 339, 354	<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition.representation.used_- representation property_definition.representation property_definition.representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition.relationship => product_definition_relationship => product_definition_usage => assembly_component_usage => (next_assembly_usage_occurrence) ([next_assembly_usage_occurrence] [quantified_assembly_component_usage]) </pre>
component_assembly_3d_position to physical_unit_3d_shape (as assembly_model)	PATH		338, 111	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape']} [representation </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition => physical_unit[]}] representation => shape_representation </pre>
COMPONENT_ASSEMBLY_- NON_PLANAR_POSITION	manifold_constraining_context_- dependent_shape_representation	210		<pre> {[manifold_constraining_context_dependent_- shape_representation <= context_dependent_shape_representation] [manifold_constraining_context_dependent_- shape_representation <= representation_relationship]]} manifold_constraining_context_dependent_- shape_representation <= [representation_relationship representation_relationship.rep_2 ->] [context_dependent_shape_representation context_dependent_shape_representation.representation_- relation -> shape_representation_relationship {shape_representation_relationship => shape_representation_relationship_with_- transformation}] </pre>
component_assembly_non_- planar_position to assembly_component_non_- planar_shape (as component_model)	PATH			

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> representation => shape_representation => manifold_subsurface_shape_representation manifold_constraining_context_dependent_ shape_representation <= context_dependent_shape_representation context_dependent_shape_representation.representation_ relation -> shape_representation_relationship => shape_representation_relationship_with_ transformation <= representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> item_defined_transformation
component_assembly_non_ planar_position to component_non_planar_location (as transformation)	PATH			
component_assembly_non_ planar_position to next_higher_assembly_ relationship (as component)	PATH			manifold_constraining_context_dependent_ shape_representation <= context_dependent_shape_representation context_dependent_shape_representation.represented_ product_relation -> product_definition_shape <= property_definition property_definition.definition -> product_definition_relationship => product_definition_usage => assembly_component_usage => next_assembly_usage_occurrence
component_assembly_non_ planar_position to physical_unit_3d_shape (as assembly_model)	PATH			manifold_constraining_context_dependent_ shape_representation <= representation_relationship representation_relationship.rep_1 -> manifold_surface_shape_representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COMPONENT_NON_PLA- NAR_LOCATION	item_defined_transformation	43		item_defined_transformation {item_defined_transformation.name = 'component non planar location'}
placement_fixed	item_defined_- transformation.description	43		item_defined_transformation {item_defined_transformation (item_defined_transformation.description = 'placement fixed true') (item_defined_transformation.description = 'placement fixed false')} item_defined_transformation.description
component_non_planar_location to axis_placement_3d (as transformation)	PATH			item_defined_transformation item_defined_transformation.transform_item_1 -> representation_item => geometric_representation_item => placement => axis2_placement_3d
COMPONENT_PART_2D_GEO- METRIC_REPRESENTATION_- RELATIONSHIP	shape_representation_relationship	41	40	{shape_representation_relationship <= representation_relationship {representation_relationship representation_relationship.name = 'component part planar shape'}}
component_part_2d_geometric_- representation_relationship to assembly_component_2d_shape (as component_shape)	PATH		340, 338, 333, 111	shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation [representation.name = 'planar projected shape'] [representation <= property_definition_representation.used_- representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> ({property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect)}} representation => shape_representation </pre>
component_part_2d_geometric_- representation_relationship to part_template_planar_shape (as part_shape)	PATH		340	<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
component_part_2d_geometric_- representation_relationship to physical_unit_planar_shape (as part_shape)	PATH		338	<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit[] representation => shape_representation
COMPONENT_PART_2D_- NON_PLANAR_GEOMETRIC_- REPRESENTATION_RELATIONSHIP	shape_representation_- relationship_with_transformation	210		{shape_representation_relationship_with_- transformation <= shape_representation_relationship <= representation_relationship {representation_relationship representation_relationship.name = 'component part non planar shape'}}}
component_part_2d_non_planar_- geometric_representation_- relationship to assembly_component_non_- planar_shape (as component_shape)	PATH			shape_representation_relationship_with_- transformation <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation [representation.name = 'non planar shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => component_definition (property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect[]}) representation => shape_representation shape_representation.relationship_with_ transformation <= shape_representation.relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[]}] </pre>
component_part_2d_non_planar_ - geometric_representation_ - relationship to part_template_planar_shape (as part_shape)	PATH			

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation => shape_representation
COMPONENT_PART_3D_GEO- METRIC_REPRESENTATION_- RELATIONSHIP	shape_representation_relationship	41		shape_representation_relationship <= representation_relationship {representation_relationship representation_relationship.name = 'component part 3d shape'}
component_part_3d_geometric_- representation_relationship to assembly_component_3d_shape (as component_shape)	PATH		41, 338, 333, 111	shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation [(representation.name = '3d bound volume shape') (representation.name = 'part template non planar 2d shape')]} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {(property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => component_definition) (property_definition property_definition.definition -> characterized_definition

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_shape_aspect[] representation => shape_representation
component_part_3d_geometric_- representation_relationship to part_template_3d_shape (as part_shape)	PATH		333, 111, 41	shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[]} representation => shape_representation
component_part_3d_geometric_- representation_relationship to part_template_non_planar_2d_- shape (as part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_ representation property_definition_representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
component_part_3d_geometric_ representation_relationship to physical_unit_3d_shape (as part_shape)	PATH		338, 111, 41	shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition =>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit[] representation => shape_representation
DESIGN_INTENT_MODIFICATION_3D_POSITION	mapped_item	43		{mapped_item <= representation_item representation_item.name = 'positioned design intent modification 3d shape'}
design_intent_modification_3d_- position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
design_intent_modification_3d_- position to physical_unit_3d_shape (as design_shape)	PATH			mapped_item <= representation_item <= representation.items[i] [[representation representation.name = '3d bound volume shape'] [representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit[]} representation => shape_representation </pre>
design_intent_modification_3d_- position to design_intent_modification_non_- planar_2d_shape (as feature_shape)	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation => manifold_surface_shape_representation </pre>
design_intent_modification_3d_- position to design_intent_modification_3d_- shape (as feature_shape)	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_- representation property_definition_representation </pre>
DESIGN_INTENT_MODIFICATION_3D_SHAPE	shape_representation	41		
design_intent_modification_3d_- shape to design_intent_modification (as represented_design_intent_- modification)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => shape_modification
DESIGN_INTENT_MODIFICATION_NON_PLANAR_2D_SHAPE	manifold_surface_shape_representation	509	290, 124	{manifold_surface_shape_representation <= shape_representation <= representation representation.name = 'design intent modification non planar 2d shape'}
design_intent_modification_non_planar_2d_shape to design_intent_modification (as represented design_intent_modification)	PATH			manifold_surface_shape_representation <= shape_representation <= representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => shape_modification
DESIGN_INTENT_MODIFICATION_PLANAR_POSITION	mapped_item	43		{mapped_item <= representation_item representation_item.name = 'design intent modification planar position'}
design_intent_modification_planar_position to physical_unit_planar_shape (as design_shape)	PATH			mapped_item <= representation_item <- representation.items[i] {representation representation.name = 'planar projected shape'} [representation <- property_definition_representation.used_representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit}} representation => shape_representation </pre>
design_intent_modification_- planar_position to axis_placement_2d (as transformation)	PATH			<pre> mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d </pre>
design_intent_modification_- planar_position to design_intent_modification_- styled_area_representation (as feature_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification_- planar_position to design_intent_modification_- planar_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
DESIGN_INTENT_MODIFICATION_PLANAR_SHAPE	shape_representation	41		
design_intent_modification_- planar_shape to design_intent_modification_- (as represented_design_intent_- modification)	PATH			shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => shape_modification
DESIGN_INTENT_MODIFICATION_STYLED_AREA_REPRESENTATION	property_definition_representation	41		
design_intent_modification_- styled_area_representation to design_intent_modification_- planar_shape (as derived_planar_shape)	PATH			property_definition_representation property_definition_representation.used_ representation -> representation <- {[representation representation.name = 'styled area representation'] [representation => shape_representation]} representation_relationship.rep_1

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship { representation_relationship.name = 'derived planar shape' } representation_relationship.rep_2 -> representation <- { representation representation.name = 'planar representation' } property_definition_representation.used_ representation property_definition_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => shape_modification </pre>
design_intent_modification_- styled_area_representation to design_intent_modification (as represented_design_intent_- modification)	PATH		333	
design_intent_modification_- styled_area_representation to fill_area_occurrence (as shape_styled_area_representation)	PATH			<pre> property_definition_representation property_definition_representation.used_ representation -> representation { representation representation.name = 'styled area representation' } representation representation => shape_representation[] </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_fill_area_occurrence </pre>
design_intent_modification_ styled_area_representation to styled_curve_occurrence (as shape_styled_area_representation)	PATH			<pre> property_definition_representation property_definition_representation.used_ representation -> representation {[representation representation.name = 'styled area representation'] [representation representation => shape_representation]} representation.items[i] -> representation_item => annotation_occurrence => annotation_curve_occurrence </pre>
DESIGN_VIEW_CONNEC- TION_ZONE_POSITION_RE- LATIONSHIP	representation_relationship_with_ transformation	43		<pre> {representation_relationship_with_ transformation representation_relationship_with_ transformation.name = 'design view connection zone position relationship'} </pre>
design_view_connection_zone_ terminal_shape_relationship to shape_definition (as associating_design_view_shape)	PATH			<pre> representation_relationship_with_ transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'design view shape'} shape_representation </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_view_connection_zone_- terminal_shape_relationship to connection_zone (as associated_usage)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => shape_representation} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => </pre>
design_view_connection_zone_- position_relationship to connection_zone (as associated_usage)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} representation_relationship_with_ transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} representation => shape_representation representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item </pre>
design_view_connection_zone_- position_relationship to shape_definition (as associated_connection_zone_- shape_definition)	PATH			
design_view_connection_zone_- position_relationship to axis_placement (as associated_usage_placement)	PATH			

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'position'} {representation => shape_representation} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit </pre>
design_view_connection_zone_- position_relationship to physical_unit_design_view (as associating_design_view)	PATH			

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_VIEW_TERMINAL_- COMPONENT_SHAPE_RELATIONSHIP	representation_relationship_with_- transformation	43		{representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'design view terminal component shape relationship'}
design_view_terminal_- component_shape_definition (as associating_component_shape)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation (representation.name = 'planar projected shape') (representation.name = '3d bound volume shape')} representation => shape_representation
design_view_terminal_- component_shape_definition to conductive_interconnect_- element_terminal (as associated_usage)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} {representation => shape_representation} representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect { shape_aspect.description = 'conductive interconnect element terminal' } representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> { representation { representation.name = 'terminal shape' } representation => shape_representation } representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item { representation_item { representation_item.name = 'origin' } representation_item => geometric_representation_item => placement } representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> { representation { representation.name = 'terminal shape' } representation => shape_representation } </pre>
design_view_terminal_- component_shape_relationship to feature_shape_definition (as associated_terminal_shape_- definition)	PATH			
design_view_terminal_- component_shape_relationship to axis_placement (as associated_usage_placement)	PATH			
design_view_terminal_- component_shape_relationship to interface_terminal (as associated_usage)	PATH			

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => component_interface_terminal </pre>
design_view_terminal_- component_shape_relationship to movable_packaged_component_- join_terminal (as associated_usage)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'terminal shape'} {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => {shape_aspect shape_aspect.description = 'movable packaged component join terminal'} component_terminal </pre>
OPEN_SHELL_BASED_SURFACE	manifold_surface_shape_representation	509		<pre> manifold_surface_shape_representation <= shape_representation <= {representation representation.name = 'open shell based surface'} representation </pre>
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		<pre> (component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect) </pre>
STRATUM_3D_SHAPE	shape_representation	41	314, 342, 343, 265, 266, 338	<pre> {shape_representation <= [representation representation.name = '3d bound volume shape'] [representation representation.context_of_items -> representation_context => geometric_representation_context space_dimension = 3] [representation <- property_definition_representation.used_ representation property_definition_representation.definition -> </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum]]} </pre>
stratum_3d_shape to stratum (as shape_characterized_stratum)	PATH		340, 265, 266, 338	<pre> shape_representation <= representation <- property_definition.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => stratum </pre>
STRATUM_AVERAGE_SUR- FACE_SHAPE	manifold_surface_shape_ representation	509	315	<pre> {manifold_surface_shape_representation <= shape_representation <= representation representation.name = 'stratum average surface shape'} </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_average_surface_shape to stratum_average_surface (as of_surface)	PATH			<pre> manifold_surface_shape_representation <= shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect {shape_aspect shape_aspect.description = 'average surface'} shape_aspect => stratum_surface </pre>
STRATUM_FEATURE_3D - SHAPE	shape_representation	41		
stratum_feature_3d_shape to stratum_feature (as represented_stratum_feature)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => stratum_feature </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
STRATUM_FEATURE_NON- PLANAR_2D_SHAPE	manifold_surface_shape_- representation	509	290, 319	{manifold_surface_shape_representation <= shape_representation <= representation.name = 'stratum feature non planar 2d shape'}
stratum_feature_non_planar_2d_- shape to stratum_feature (as represented_stratum_feature)	PATH			manifold_surface_shape_representation <= shape_representation <= representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => stratum_feature
STRATUM_FEATURE_PLA- NAR_SHAPE	shape_representation	41		
stratum_feature_planar_shape to stratum_feature (as represented_stratum_feature)	PATH			shape_representation <= representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => stratum_feature
STRATUM_FEATURE_- SHAPE_STRATUM_AVERAGE_SURFACE_SHAPE_RELATIONSHIP	representation_relationship	43	320	{representation_relationship representation_relationship.name = 'stratum feature shape stratum average surface shape relationship'}
stratum_feature_shape_stratum_- average_surface_shape_- relationship to stratum_feature_non_planar_2d_- shape (as of_stratum_feature_shape)	PATH			representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'stratum feature non planar 2d shape'}
stratum_feature_shape_stratum_- average_surface_shape_- relationship to stratum_average_surface_shape (as of_stratum_average_surface_- shape)	PATH			representation => shape_representation => manifold_surface_shape_representation representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'stratum average surface shape'}
				representation => shape_representation => manifold_surface_shape_representation
STRATUM_FEATURE_- SHAPE_STRATUM_SURFACE_SHAPE_RELATIONSHIP	representation_relationship	43	321	{representation_relationship representation_relationship.name = 'stratum feature shape stratum surface shape relationship'}
stratum_feature_shape_stratum_- surface_shape_relationship to stratum_feature_non_planar_2d_- shape (as of_stratum_feature_shape)	PATH			representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'stratum feature non planar 2d shape'}
				representation => shape_representation => manifold_surface_shape_representation

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_feature_shape_stratum_- surface_shape_relationship to stratum_surface_shape (as of_stratum_surface_shape)	PATH			representation_relationship representation {representation.name = 'stratum surface shape'} representation => shape_representation => manifold_surface_shape_representation
STRATUM_FEATURE_- STYLED_AREA_REPRESENTATION	property_definition_representation	41		
stratum_feature_styled_area_- representation to stratum_feature_planar_shape (as derived_planar_shape)	PATH			property_definition_representation property_definition_representation.used_- representation -> {representation representation.name = 'styled area representation'} representation => shape_representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'derived planar shape'} representation_relationship.rep_2 -> representation {representation.name = 'planar representation'} representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition
stratum_feature_styled_area_- representation to stratum_feature (as represented_stratum_feature)	PATH		333	

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition shape_definition = shape_aspect shape_aspect => stratum_feature
stratum_feature_styled_area_- representation to fill_area_occurrence (as shape_styled_area_representation)	PATH			property_definition_representation property_definition_representation.used_- representation -> {[representation representation.name = 'styled area representation'] [representation representation => shape_representation]} representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_fill_area_occurrence
stratum_feature_styled_area_- representation to styled_curve_occurrence (as shape_styled_area_representation)	PATH			property_definition_representation property_definition_representation.used_- representation -> {[representation representation.name = 'styled area representation'] [representation representation => shape_representation]} shape_representation.items[i] ->

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence </pre>
STRATUM_PLANAR_SHAPE	shape_representation	41	326, 342, 265, 266, 338	<pre> {shape_representation <= [representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => stratum]]} </pre>
stratum_planar_shape to stratum (as shape_characterized_stratum)	PATH		340, 265, 266, 338	<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum </pre>
STRATUM_SURFACE_IN_- STRATUM_3D_POSITION	mapped_item	43	327	<pre> {mapped_item <= representation_item representation_item.name = 'stratum surface in stratum 3d position'} </pre>
stratum_surface_in_stratum_3d_- position to stratum_3d_shape (as stratum_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 65 – Mapping table for interconnect_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum[] representation => shape_representation
stratum_surface_in_stratum_3d_ position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
stratum_surface_in_stratum_3d_ position to stratum_surface_shape (as surface_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {representation representation.name = 'stratum surface shape'} representation => shape_representation => manifold_surface_shape_representation
stratum_surface_in_stratum_3d_ position to stratum_average_surface_shape (as surface_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {representation representation.name = 'stratum average surface shape'} representation =>

Table 65 – Mapping table for interconnect_shape UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => manifold_surface_shape_representation
STRATUM_SURFACE_SHAPE	manifold_surface_shape_- representation	509	328	{manifold_surface_shape_representation <= shape_representation <= representation representation.name = 'stratum surface shape'}
stratum_surface_shape to stratum_surface (as of_surface)	PATH			manifold_surface_shape_representation <= shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect (shape_aspect.description = 'primary surface') (shape_aspect.description = 'secondary surface')} shape_aspect => stratum_surface

Table 66 – Mapping table for laminate_assembly_design UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_JOINT	assembly_joint	210		assembly_joint <=[shape_aspect_relationship][shape_aspect]
description	shape_aspect_relationship.description	41	344	assembly_joint <=[shape_aspect_relationship][shape_aspect_relationship.description]
name	shape_aspect.name	41	333	assembly_joint <=[shape_aspect][shape_aspect.name]
assembly_joint to assembly_group_component (as associated_assembly)	PATH			assembly_joint <=[shape_aspect][shape_aspect.of_shape -> product_definition_shape <=[property_definition][property_definition.definition -> characterized_definition -> product_definition][characterized_product_definition = product_definition][product_definition => assembly_group_component_definition]
assembly_joint to assembly_module (as associated_assembly)	PATH			assembly_joint <=[shape_aspect.of_shape -> product_definition_shape <=[property_definition.definition -> characterized_definition -> product_definition][characterized_product_definition = product_definition]

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition {product_definition.name = 'assembly module'}
assembly_joint to component_mounting_feature (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => component_mounting_feature
assembly_joint to component_mounting_feature (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => component_mounting_feature
assembly_joint to interconnect_- component_interface_terminal (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect component interface terminal'} shape_aspect => component_interface_terminal
assembly_joint to interconnect_- component_interface_terminal (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interconnect component interface terminal'} shape_aspect => component_interface_terminal
assembly_joint to interconnect_module_- component_surface_feature (as assembly_features_1)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => interconnect_module_component_surface_feature

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_joint to interconnect_module - component_surface_feature (as assembly_features_2)	PATH			assembly_joint <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => interconnect_module.component_surface_feature
COMPONENT_MATING_CON- STRAINT_CONDITION	component_mating_constraint_ condition	210		component_mating_constraint_condition <= shape_aspect
constrained_degree_of_freedom	representation_item.name	43		<pre> component_mating_constraint_condition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation property_definition_representation.description = 'constrained degrees of freedom'} property_definition_representation.used_ representation -> representation {representation representation.name = 'degrees of freedom'} representation.items[i] -> {(representation_item.name = 'x') (representation_item.name = 'y') (representation_item.name = 'z') (representation_item.name = 'u') (representation_item.name = 'v') (representation_item.name = 'w')}} </pre>

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
name	shape_aspect.name	41		component_mating_constraint_condition <= shape_aspect shape_aspect.name
component_mating_constraint_- condition to assembly_joint (as constraining_joint)	PATH			component_mating_constraint_condition <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'constraining_joint'}
component_mating_constraint_- condition to physical_unit_design_view (as of_context)	PATH			shape_aspect_relationship.related_shape_aspect -> shape_aspect => assembly_joint component_mating_constraint_condition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'}

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERCONNECT_MODULE	physical_unit	210	338, 268	<pre> physical_unit <= product_definition {product_definition.name = 'interconnect module'} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] </pre>
interconnect_module to assembly_module (as assembly_design_requirement)	PATH			<pre> physical_unit <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship.name = product_definition_relationship.name = 'assembly_design_requirement'} product_definition_relationship.relatin_ - product_definition -> product_definition {product_definition => [product_definition => physical_unit] [product_definition.name = 'assembly module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']] </pre>
INTERCONNECT_MODULE_- COMPONENT	component_definition	210	338, 268	<pre> component_definition <= product_definition {product_definition [product_definition product_definition.description != 'laminate component']} </pre>

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']] component_definition <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'instantiated part'} product_definition_relationship product_definition_relationship.relatin_ - product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)</pre>
interconnect_module_component to interconnect_module_usage_view (as usage_definition)	PATH		338, 268	
INTERCONNECT_MODULE_- COMPONENT_SURFACE_FEA- TURE	interconnect_module_- component_surface_feature	210	333	<pre>interconnect_module_component_surface_feature <= shape_aspect {shape_aspect.description = 'interconnect module component surface feature'}</pre>

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_- component_surface_feature to interconnect_module_surface_ feature (as definition)	PATH			interconnect_module_component_surface_feature <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'}} shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}}}
NEXT_HIGHER_ASSEMBLY_- RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44	213, 214	(next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=] assembly_component_usage)

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
reference_designation	PATH			next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator
next_higher_assembly_- relationship to data_element (as quantity)	PATH			next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage -> measure_with_unit
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PROMISSORY_USAGE_IN_- PRODUCT_MODEL_RELAT- TIONSHIP	applied_promissory_usage_in_- product_model_assignment	210		{applied_promissory_usage_in_product_model_- assignment <= group_assignment group_assignment.assigned_group -> group => promissory_usage_in_product_model}
promissory_usage_in_product_- model_relationship to Ee_product_definition (as component)	PATH			applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_definition (product_definition => product_definition_with_associated_documents)

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (product_definition) (product_definition => evaluation_product_definition) (product_definition => physical_unit) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition => analytical_model_definition) (product_definition => footprint_definition) (product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => stratum) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit) applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_concept </pre>
promissory_usage_in_product_- model_relationship to ee_product_model (as product_model)	PATH			

Table 66 – Mapping table for laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
promissory_usage_in_product_model_relationship to security_classification (as security_code)	PATH			<pre> applied_promissory_usage_in_product_model_assignment security_classification_assigned_item = applied_promissory_usage_in_product_model_assignment security_classification_assigned_item <- applied_security_classification_assignment.items[1] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_security_classification -> security_classification </pre>
PROMISSORY_USAGE_RELATIONSHIP	promissory_usage_occurrence	44	275, 276	
promissory_usage_relationship to physical_unit (as component)	PATH		354, 339, 338	<pre> promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design')} (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>

Table 66 – Mapping table for laminate_assembly_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
promissory_usage_relationship to physical_unit_design_view (as assembly)	PATH			<pre> promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatiing_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit </pre>
promissory_usage_relationship to security_classification (as security_code)	PATH			<pre> promissory_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_ assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>

Table 67 – Mapping table for land UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_LAND	land	210	340, 338, 268	<pre> land <= component_shape_aspect <= shape_aspect {shape_aspect {shape_aspect.description = 'component termination passage and contact size dependent land'}} [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition.description = 'laminated component']] </pre>
component_termination_passage_and_contact_size_dependent_land to component_termination_passage (as reference_passage)	PATH		333	<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference passage'} shape_aspect_relationship.related_shape_ aspect -> </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
<pre> component_termination_passage_- and_contact_size_dependent_land to_default_attachment_size_and_- component_termination_passage_- based_land_physical_template (as usage_definition) </pre>	PATH		333, 340	<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default attachment size and component termination passage based'} shape_aspect => part_template_definition => land_physical_template </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND	land	210	333, 340, 338, 268	<pre> land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'component termination passage and contact size dependent non functional land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']]}</pre>
component_termination_passage_- and_contact_size_dependent_- non_functional_land to component_termination_passage (as reference_passage)	PATH		333	<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference passage'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect}</pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.description = 'component termination passage' shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
<pre> component_termination_passage_- and_contact_size_dependent_- non_functional_land to default_attachment_size_and_- component_termination_passage_- based_land_physical_template (as usage_definition) </pre>	PATH			<pre> Land <= component_shape_aspect <= shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default attachment size and component termination passage based'} shape_aspect => part_template_definition => land_physical_template </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
COMPONENT_TERMINATION_PASSAGE_DEPENDENT_LAND	land	210	333, 340, 338, 268	land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'component termination passage dependent land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'lamine component']}]}
component_termination_passage_ dependent_land to component_termination_passage (as reference_passage)	PATH		333	land <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'reference passage'}}

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_ aspect -> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
component_termination_passage_- dependent_land to default_component_termination_- passage_based_land_physical_- template (as usage_definition)	PATH			<pre> Land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_ aspect -> {shape_aspect shape_aspect.description = 'default component termination passage based'} shape_aspect => part_template_definition => land_physical_template </pre>
COMPONENT_TERMINATION_PASSAGE_DEPENDENT_NON_FUNCTIONAL_-LAND	land	210	333, 340, 338, 268	<pre> Land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'component termination passage dependent non functional land']} [shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition.description = 'laminated component']}] </pre>
component_termination_passage_- dependent_non_functional_land to component_termination_passage (as reference_passage)	PATH		333	<pre> Land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference passage'} shape_aspect_relationship.related_shape_ aspect -> {shape_aspect shape_aspect.description = 'component termination passage'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
component_termination_passage_- dependent_non_functional_land to default_component_termination_- passage_based_land_physical_- template (as usage_definition)	PATH			<pre> Land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_ aspect -> {shape_aspect shape_aspect.description = 'default component termination passage based'} </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => part_template_definition => land_physical_template
CONTACT_SIZE_DEPENDENT_LAND	land	210	333, 340, 338, 268	land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'contact size dependent land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']}]
contact_size_dependent_land to default_attachment_size_based_ land_physical_template (as usage_definition)	PATH		333	land <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_ aspect ->

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect shape_aspect.description = 'default attachment size based'} shape_aspect => part_template_definition => land_physical_template</pre>
CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_-LAND	land	210	340, 338, 268	<pre>land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'contact size dependent non functional land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']]}</pre>
contact_size_dependent_non_-functional_land to default_attachment_size_based_-land_physical_template (as usage_definition)	PATH		333	<pre>land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect</pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.description = 'default attachment size based' shape_aspect => part_template_definition => land_physical_template </pre>
LAND	land	210	340, 338, 268	<pre> land <= component_shape_aspect <= shape_aspect {[(shape_aspect.description = 'functional land') (shape_aspect.description = 'via dependent land') (shape_aspect.description = 'component termination passage dependent land') (shape_aspect.description = 'contact size dependent land') (shape_aspect.description = 'component termination passage and contact size dependent land') (shape_aspect.description = 'via and contact size dependent land')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition [product_definition => component_definition] [product_definition product_definition.definition = 'laminate component']] </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
land to land_physical_template (as alternate_land_definition)	PATH		333	<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'alternate instantiated template'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => part_template_definition => land_physical_template </pre>
land to land_physical_template (as usage_definition)	PATH			<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect (shape_aspect.description = 'stratum feature template') (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based') (shape_aspect.description = 'default unsupported passage based')} </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'default component termination passage based') shape_aspect => part_template_definition
NON_FUNCTIONAL_LAND	land	210	340, 338, 268	land <= component_shape_aspect <= shape_aspect {shape_aspect [(shape_aspect.description = 'non functional land') (shape_aspect.description = 'via dependent non functional land') (shape_aspect.description = 'component termination passage dependent non functional land') (shape_aspect.description = 'contact size dependent non functional land') (shape_aspect.description = 'unsupported passage dependent non functional land') (shape_aspect.description = 'via and contact size dependent non functional land')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition]

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
non_functional_land to land_physical_template (as alternate_land_definition)	PATH		333	<pre>[product_definition product_definition.description = 'laminated component']] land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'alternate instantiated template'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => part_template_definition => land_physical_template</pre>
non_functional_land to land_physical_template (as usage_definition)	PATH			<pre>land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect shape_aspect.description = 'stratum feature template'} shape_aspect => part_template_definition => land_physical_template</pre>
UNSUPPORTED_PASSAGE_- DEPENDENT_NON_FUNC- TIONAL_LAND	land	210	340, 338, 268	<pre>land <= component_shape_aspect <= shape_aspect {shape_aspect shape_aspect.description = 'unsupported passage dependent non functional land'} shape_aspect.of_shape -></pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']] </pre>
unsupported_passage_dependent_- non_functional_land to unsupported_passage (as reference_passage)	PATH		333	<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference passage'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'unsupported passage'} shape_aspect => component_shape_aspect </pre>
unsupported_passage_dependent_- non_functional_land to default_unsupported_passage_- based_land_physical_template (as usage_definition)	PATH			<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.description = 'default unsupported passage based' shape_aspect => part_template_definition => land_physical_template </pre>
VIA AND CONTACT_SIZE_- DEPENDENT_LAND	land	210	340, 338, 268	<pre> land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'via and contact size dependent land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'lamineate component']] </pre>
via_and_contact_size_dependent_- land to via (as reference_via)	PATH			<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'reference via' shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'via'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
<pre> via_and_contact_size_dependent_- land to default_attachment_size_and_- via_based_land_physical_template (as usage_definition) </pre>	PATH			<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default attachment size and via based'} </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>shape_aspect =></code> <code>part_template_definition =></code> <code>land_physical_template</code>
VIA_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND	land	210	333, 340, 338, 268	<code>land <=</code> <code>component_shape_aspect <=</code> <code>shape_aspect</code> <code>{shape_aspect</code> <code>[shape_aspect.description = 'via and contact</code> <code>size dependent non functional land']</code> <code>[shape_aspect.of_shape -></code> <code>product_definition_shape <=</code> <code>property_definition</code> <code>property_definition.definition -></code> <code>characterized_definition</code> <code>characterized_definition = characterized_</code> <code>product_definition</code> <code>characterized_product_definition = product_</code> <code>definition</code> <code>[product_definition =></code> <code>component_definition]</code> <code>[product_definition</code> <code>product_definition.description = 'lamine</code> <code>component']}]}</code>
via_and_contact_size_dependent_non_functional_land to via (as reference_via)	PATH			<code>land <=</code> <code>component_shape_aspect <=</code> <code>shape_aspect <-</code> <code>shape_aspect_relationship.related_shape_aspect</code> <code>shape_aspect_relationship</code> <code>{shape_aspect_relationship</code> <code>shape_aspect_relationship.name = 'reference</code> <code>via'}</code>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'via'} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage
via_and_contact_size_dependent_- non_functional_land to default_attachment_size_and_- via_based_land_physical_template (as usage_definition)	PATH			land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default attachment size and via based'} shape_aspect =>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				part_template_definition => land_physical_template
VIA_DEPENDENT_LAND	land	210	333, 340, 338, 268	land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'via dependent land'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']}]}
via_dependent_land to default_ via_based_land_physical_template (as usage_definition)	PATH		333	land <= component_shape_aspect <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'}

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default via based'} shape_aspect => part_template_definition => land_physical_template </pre>
via_dependent_land to via (as reference_via)	PATH			<pre> land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference via'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage </pre>
VIA_DEPENDENT_NON_- FUNCTIONAL_LAND	land	210	333, 340, 338, 268	<pre> land <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'via dependent non functional land']} </pre>

Table 67 – Mapping table for land UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component']]}</pre>
via_dependent_non_functional_- land to default_via_based_land_- physical_template (as usage_definition)	PATH		333	<pre>land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated template'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'default via based'} shape_aspect => part_template_definition => land_physical_template</pre>
via_dependent_non_functional_- land to via (as reference_via)	PATH			<pre>land <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship</pre>

Table 67 – Mapping table for land UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>shape_aspect_relationship.name = 'reference via' shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'bonded conductive base blind via') (shape_aspect.description = 'buried via') (shape_aspect.description = 'interfacial connection') (shape_aspect.description = 'non conductive base blind via') (shape_aspect.description = 'plated conductive base blind via')} shape_aspect => component_shape_aspect => inter_stratum_feature => plated_inter_stratum_feature => plated_passage</pre>

Table 68 – Mapping table for library_item UoF

Application element	AIM element	Source	Rules	Reference path
LIBRARY_EE_PRODUCT_DEFINITION	(library_defined_physical_unit) (library_defined_functional_unit)	210 210		(library_defined_physical_unit <= [physical_unit <= product_definition]) (library_defined_functional_unit <= externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition])

Table 69 – Mapping table for manufacturing_technology UoF

Application element	AIM element	Source	Rules	Reference path
COMPONENT_TERMINATION_PASSAGE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= shape_aspect {shape_aspect shape_aspect.description = 'component termination passage template'}
component_termination_passage_- template to default_component_termination_- passage_definition (as of_passage_technology)	PATH		333	shape_aspect shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'default component termination passage definition'} shape_aspect => passage_technology
COUNTERBORE_PASSAGE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'counterbore passage template'}
counterbore_passage_template to unsupported_passage_template (as smaller_passage)	PATH		333, 340	part_template_definition <= shape_aspect shape_aspect <= shape_aspect_relationship.related_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'smaller passage'}

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'larger passage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
counterbore_passage_template to blind_passage_template (as larger_passage)	PATH			
COUNTERSUNK_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect shape_aspect.description = 'countersunk passage template'} part_template_definition <= shape_aspect shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship.description = 'constant diameter passage'} </pre>
countersunk_passage_template to unsupported_passage_template (as constant_diameter_passage)	PATH		333, 340	

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition part_template_definition <= shape_aspect shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect shape_aspect.relationship {shape_aspect.relationship.description = 'larger passage'} shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'unsupported passage template'} shape_aspect => part_template_definition </pre>
countersunk_passage_template to blind_passage_template (as tapered_passage)	PATH			
DEFAULT_ATTACHMENT_- SIZE_AND_COMPONENT_- TERMINATION_PASSAGE_- BASED_LAND_PHYSICAL_- TEMPLATE	land_physical_template	210		<pre> land_physical_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'default attachment size and component termination passage based'} </pre>
DEFAULT_ATTACHMENT_- SIZE_AND_VIA_BASED_- LAND_PHYSICAL_TEMPLATE	land_physical_template	210		<pre> land_physical_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'default attachment size and via based'} </pre>
DEFAULT_ATTACHMENT_- SIZE_BASED_LAND_PHYSI- CAL_TEMPLATE	land_physical_template	210	333, 340	<pre> land_physical_template <= part_template_definition <= {shape_aspect (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based')} </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
default_attachment_size_based_- land_physical_template to connection_zone (as maximum_attachment_- region_size)	PATH			land_physical_template <= part_template_definition <= shape_aspect <- shape_aspect_relationship.relatiing_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'maximum attachment region size'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'connection zone'}
default_attachment_size_based_- land_physical_template to connection_zone (as minimum_attachment_- region_size)	PATH			land_physical_template <= part_template_definition <= shape_aspect <- shape_aspect_relationship.relatiing_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'minimum attachment region size'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'connection zone'}
DEFAULT_COMPONENT_- TERMINATION_PASSAGE_- BASED_LAND_PHYSICAL_- TEMPLATE	land_physical_template	210	333, 340	land_physical_template <= part_template_definition <= {shape_aspect (shape_aspect.description = 'default component termination passage based') (shape_aspect.description = 'default attachment size and component termination passage based')}

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
default_component_termination_- passage_based_land_physical_- template to default_component_termination_- passage_definition (as of_passage_technology)	PATH		333, 340	land_physical_template <= part_template_definition <= shape_aspect <= shape_aspect_relationship.relatiing_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'technology usage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect shape_aspect.description = 'default component termination passage definition'} passage_technology
DEFAULT_COMPONENT_TERMINATION	passage_technology	210		passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default component termination passage definition'}
default_component_termination_- passage_definition to ee_material (as deposition_material)	PATH			passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default component termination passage definition'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= material_designation.definitions[i] material_designation
default_component_termination_- passage_definition to length_data_element (as maximum_allowed_- component_terminal_extent)	PATH		333, 109	passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <=

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'physical characteristics representation'} representation representation.items[i] -> {representation_item representation_item.name = 'maximum allowed component terminal extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
<pre> default_component_termination_- passage_definition to length_data_element (as minimum_allowed_- component_terminal_extent) </pre>	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'physical characteristics representation'} representation representation.items[i] -> {representation_item </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item.name = 'minimum allowed component terminal extent' representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
<pre> default_component_termination_- passage_definition to length_data_element (as minimum_as_finished_- deposition_thickness) </pre>	PATH		333, 109	<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> {representation representation.name = 'physical characteristics representation'} representation representation.items[i] -> {representation_item representation_item.name = 'minimum as finished deposition thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
<pre> default_component_termination_- passage_definition to length_data_element (as minimum_as_finished_- passage_extent) </pre>	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'physical characteristics representation'} representation representation.items[i] -> {representation_item representation_item.name = 'minimum as finished passage extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
DEFAULT_LAND_PHYSICAL_ TEMPLATE	land_physical_template	210	333, 340	<pre> land_physical_template <= part_template_definition <= {shape_aspect (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based') (shape_aspect.description = 'default unsupported passage based') (shape_aspect.description = 'default component termination passage based')} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
default_land_physical_template to stratum_technology (as of_stratum_technology)	PATH		333	land_physical_template <= part_template_definition <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ object characterized_object => {characterized_object.description = 'design layer'} stratum_technology
DEFAULT_PASSAGE_BASED_ LAND_PHYSICAL_TEMPLATE	land_physical_template	210	333, 340, 95	land_physical_template <= part_template_definition <= {shape_aspect (shape_aspect.description = 'default via based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default unsupported passage based')}

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'default component termination passage based') (shape_aspect.description = 'default attachment size and component termination passage based')}
default_passage_based_land_- physical_template to inter_stratum_extent (as of_stratum_extent)	PATH			<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'of stratum extent'} property_definition.relationship.relatin_- property_definition -> property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'} </pre>
default_passage_based_land_- physical_template to passage_technology (as of_passage_technology)	PATH		333	<pre> shape_aspect <- shape_aspect_relationship.relatin_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'technology usage'} shape_aspect_relationship.related_shape_aspect -> </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect => passage_technology
default_passage_based_land_- physical_template to length_data_element (as minimum_fabrication_allowance)	PATH		109	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition property_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum fabrication_allowance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
default_passage_based_land_- physical_template to length_data_element (as minimum_annular_ring)	PATH		109	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item.name = 'minimum annular ring' representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
DEFAULT_UNSUPPORTED_- PASSAGE_BASED_LAND_- PHYSICAL_TEMPLATE	land_physical_template	210	333, 340	<pre> land_physical_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'default unsupported passage based'} </pre>
default_unsupported_passage_- based_land_physical_template to default_unsupported_passage_- definition (as of_passage_technology)	PATH		333	<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'technology usage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'default unsupported passage definition'} shape_aspect => passage_technology </pre>
DEFAULT_UNSUPPORTED_- PASSAGE_DEFINITION	passage_technology	210	333	<pre> passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default unsupported passage definition'} </pre>
default_unsupported_passage_- definition to length_data_element (as minimum_as_finished_- passage_extent)	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum as finished passage extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
DEFAULT_VIA_BASED_- LAND_PHYSICAL_TEMPLATE	land_physical_template	210	333, 340	<pre> land_physical_template <= part_template_definition <= {shape_aspect (shape_aspect.description = 'default via based')} (shape_aspect.description = 'default attachment size and via based')}} </pre>
default_via_based_land_- physical_template to default_via_definition (as of_passage_technology)	PATH		333	<pre> shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'technology usage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.description = 'default via definition'} shape_aspect => passage_technology
DEFAULT_VIA_DEFINITION	passage_technology	210		passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default via definition'}
default_via_definition to ee_material (as deposition_material)	PATH		333	passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default via definition'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- material_designation.definitions[i] material_designation
default_via_definition to length_data_element (as minimum_as_finished_- deposition_thickness)	PATH		109	passage_technology <= shape_aspect {shape_aspect shape_aspect.description = 'default via definition'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation ->

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation representation.items[i] -> {representation_item representation_item.name = 'minimum as finished deposition thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
default_via_definition to length_data_element (as minimum_as_finished_- passage_extent)	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum as finished passage extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
DEPENDENT_MATERIAL_RE- MOVAL_FEATURE_TEMPLATE	dependent_material_removal_- feature_template	210		<pre> dependent_material_removal_feature_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'material removal feature template'} </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
dependent_material_removal_- feature_template to default_land_physical_template (as associated_template)	PATH			<pre> dependent_material_removal_feature_template <= part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated template'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {[shape_aspect => part_template_definition => land_physical_template] [shape_aspect (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based') (shape_aspect.description = 'default unsupported passage based') (shape_aspect.description = 'default component termination passage based')]]} </pre>
DESIGN_LAYER_TECHNOL- OGY	stratum_technology	210		<pre> stratum_technology <= characterized_object {characterized_object characterized_object.description = 'design layer'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_layer_purpose	descriptive_representation_item	45		<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'layer purpose'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'power or ground') (descriptive_representation_item.description = 'other signal') (descriptive_representation_item.description = 'lands only')} </pre>
design_layer_technology to length_data_element (as minimum_finished_feature_size)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature size'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
<pre> design_layer_technology to length_data_element (as minimum_finished_feature_ spacing) </pre>	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature spacing'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
DOCUMENTATION_LAYER_- TECHNOLOGY	stratum_technology	210		stratum_technology <= characterized_object {characterized_object characterized_object.description = 'documentation_layer/'}
documentation_layer_purpose	descriptive_representation_- item.description	45	109	stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> representation_item representation_item.name = 'layer purpose'} representation_item => descriptive_representation_item.description
FABRICATION_TECHNOL- OGY_SPECIFICATION	ee_specification	210		ee_specification <= document {document

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document.kind -> document_type document_type.product_data_type = 'fabrication technology specification'}
INTER_STRATUM_FEATURE_- TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'via template') (shape_aspect.description = 'component termination passage template') (shape_aspect.description = 'unsupported passage template') (shape_aspect.description = 'inter stratum feature edge template') (shape_aspect.description = 'inter stratum feature template') (shape_aspect.description = 'counterbore passage template') (shape_aspect.description = 'countersunk passage template') (shape_aspect.description = 'blind passage template')}}}
inter_stratum_feature_template to passage_technology (as of_passage_technology)	PATH			shape_aspect shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => passage_technology
LAND_PHYSICAL_TEMPLATE	land_physical_template	210	333, 340, 200	land_physical_template <= part_template_definition <= shape_aspect

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
land_physical_template to external_definition (as pre_defined_classification_code)	PATH			land_physical_template <= part_template_definition document_assigned_item = part_template_definition document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'predefined_classification code'}
LAND_TEMPLATE_INTERFACE_TERMINAL	land_template_terminal	210	333	land_template_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interface terminal'}
land_template_interface_terminal to default_attachment_size_-based_land_physical_template (as associated_definition)	PATH			land_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated definition'} shape_aspect_relationship.relatng_shape_-aspect -> shape_aspect =>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based')} part_template_definition => land_physical_template</pre>
LAND_TEMPLATE_JOIN_TERMINAL	land_template_terminal	210		<pre>land_template_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'join terminal'}</pre>
LAND_TEMPLATE_TERMINAL	land_template_terminal	210	333, 201	<pre>land_template_terminal <= shape_aspect {[shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')]} [shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition product_definition.formation -> product_definition_formation</pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model' [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'[]}] </pre>
name	shape_aspect.name	41	333	<pre> land_template_terminal <= shape_aspect shape_aspect.name </pre>
connection_zone_category	shape_aspect- relationship.description	41	333	<pre> land_template_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship shape_aspect_relationship.description {(shape_aspect_relationship.description = 'edge curve') (shape_aspect_relationship.description = 'edge point') (shape_aspect_relationship.description = 'surface area') (shape_aspect_relationship.description = 'surface point')} </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
land_template_terminal to connection_zone (as terminal_connection_zone)	PATH		333	land_template_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
land_template_terminal to land_physical_template (as associated_definition)	PATH		340, 333	land_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated definition'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_template_definition => land_physical_template
LINE_WIDTH_CLASS	styled_item	46		
class_name	representation_item.name	43	109	styled_item <= representation_item representation_item.name
line_width_class to curve_style (as class_member)	PATH		70	styled_item <= representation_item <- representation.items[i] representation {representation representation.name = 'curve style parameters'}

Table 69 – Mapping table for manufacturing_ technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PASSAGE_TECHNOLOGY	passage_technology	210	333, 340	<pre> passage_technology <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition = product_- definition product_definition [product_definition product_definition.name = 'interconnect module']} </pre>
maximum_aspect_ratio	ratio_measure_with_unit	41	333, 109	<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum aspect ratio'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => measure_representation_item <= measure_with_unit => ratio_measure_with_unit <= measure_with_unit.value_component -> measure_value measure_value = positive_ratio_measure positive_ratio_measure </pre>
name	shape_aspect.name	41	333	<pre> passage_technology <= shape_aspect shape_aspect.name </pre>
plated_passage	descriptive_representation_item	45		<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'plated passage'} representation_item => descriptive_representation_item {descriptive_representation_item </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
passage_technology to ee_material (as deposition_material)	PATH		333	passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= material_designation.definitions[i] material_designation
passage_technology to inter_stratum_extent (as as_finished_inter_stratum_extent)	PATH		333, 340	passage_technology <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'as finished inter_stratum extent'} shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition property_definition.description = 'finished stratum extent'} property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'}</pre>
passage technology to interconnect module (as associated_definition)	PATH			<pre>passage technology <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition {product_definition.name = 'interconnect module'}</pre>
passage technology to length_data_element (as maximum_as_finished_- passage_extent)	PATH		333, 109	<pre>passage technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation.name = 'physical characteristics representation'}</pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> {representation_item representation_item.name = 'maximum as finished passage extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
passage_technology to length_data_element (as minimum_as_finished_- passage_extent)	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum as finished passage extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
passage_technology to length_data_element (as maximum_as_finished_- deposition_thickness)	PATH		333, 109	<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum as finished deposition thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
<pre> passage_technology to length_data_element (as minimum_as_finished_ deposition_thickness) </pre>	PATH			<pre> passage_technology <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> {representation_item representation_item.name = 'minimum as finished deposition thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
passage technology to process_specification (as specification)	PATH			<pre> passage_technology <= shape_aspect document_assigned_item = shape_aspect document_assigned_item <- applied_document_reference.items[i] applied_document_reference applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'process specification'} document => ee_specification </pre>
STRATUM_TECHNOLOGY	stratum_technology	210		<pre> stratum_technology <= characterized_object </pre>
laminare_stiffness_class	descriptive_representation_item	45	109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation' representation.items[i] -> {representation_item representation_item.name = 'laminare stiffness class'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'fluid like') (descriptive_representation_item.description = 'conformal coat') (descriptive_representation_item.description = 'stiff laminate')}} </pre>
name	characterized_object.name	41		<pre> stratum_technology <= characterized_object characterized_object.name </pre>
layer_position	descriptive_representation_item	45		<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.name = 'physical characteristics representation' } representation.items[i] -> { representation_item representation_item.name = 'layer position' representation_item => { descriptive_representation_item { descriptive_representation_item (descriptive_representation_item.description = 'secondary') } (descriptive_representation_item.description = 'all') (descriptive_representation_item.description = 'external') (descriptive_representation_item.description = 'primary') (descriptive_representation_item.description = 'internal') } } </pre>
stratum_technology to ee_material (as stratum_material)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- material_designation.definitions[i] material_designation </pre>
stratum_technology to length_data_element (as maximum_feature_size_- requirement)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum feature size requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as maximum_stratum_thickness)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum thickness'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_stratum_thickness)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_finished_feature_size)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- </pre>

Table 69 – Mapping table for manufacturing technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature size'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_finished_feature_ - spacing)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature spacing'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit stratum_technology <= characterized_object document_assigned_item = characterized_object document_assigned_item <- applied_document_reference.items[i] applied_document_reference applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'process specification'} document => ee_specification </pre>
stratum_technology to process_specification (as specification)	PATH			
UNSUPPORTED_PASSAGE_- TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect (shape_aspect.description = 'unsupported passage_template') (shape_aspect.description = 'counterbore passage_template') (shape_aspect.description = 'countersunk passage_template') (shape_aspect.description = 'blind passage template')} </pre>
unsupported_passage_template to default_unsupported_passage_- definition (as of_passage_technology)	PATH		333	<pre> shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} </pre>

Table 69 – Mapping table for manufacturing_technology UoF (Concluded)

Application element	AIMI element	Source	Rules	Reference path
				shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default unsupported passage definition'} shape_aspect => passage_technology
VIA_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'via template'}
via_template to default_via_definition (as of_passage_technology)	PATH		333	shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'inter stratum feature passage technology'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'default via definition'} shape_aspect => passage_technology

Table 70 – Mapping table for material_product UoF

Application element	AIM element	Source	Rules	Reference path
EE_MATERIAL	material_designation	45	151, 152	
electrical_conductivity_classification	material_electrical_conductivity_category	210		<pre> material_designation classification_assigned_item = material_ designation classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> {group (group.name = 'conductive')} (group.name = 'non conductive')} (group.name = 'resistive')} (group.name = 'semi conductive')}} group => material_electrical_conductivity_category </pre>
material_name	material_designation.name	45		
optical_conductivity_classification	material_optical_conductivity_category	210		<pre> material_designation classification_assigned_item = material_ designation classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> {group (group.name = 'conductive')} (group.name = 'non conductive')} </pre>

Table 70 – Mapping table for material_product UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(group.name = 'resistive') (group.name = 'semi conductive')}} group => material_optical_conductivity_category</pre>
thermal_conductivity_classification	material_thermal_conductivity_category	210		<pre>material_designation classification_assigned_item = material_ classification classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> {group (group.name = 'conductive') (group.name = 'non conductive') (group.name = 'resistive') (group.name = 'semi conductive')}} group => material_thermal_conductivity_category</pre>
ee_material to ee_document (as registration_document)	PATH			<pre>material_designation document_assigned_item = material_designation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document</pre>
EE_PRODUCT_CATEGORY	characterized_product_category	210		<pre>characterized_product_category characterized_product_category <= [product_category] [characterized_object]</pre>
description	product_category.description	41		<pre>characterized_product_category <= product_category product_category.description</pre>

Table 70 – Mapping table for material_product UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
name	product_category.name	41		characterized_product_category <= product_category product_category.name
ee_product_category to model_parameter (as category_parameter)	PATH		340, 109	characterized_product_category <= characterized_object characterized_definition = characterized_ object characterized_definition characterized_definition <= property_definition.definition property_definition {property_definition.description = 'product category parameter'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => model_parameter
ee_product_category to parameter_assignment (as category_specific_values)	PATH			characterized_product_category <= characterized_object characterized_definition = characterized_ object characterized_definition characterized_definition <= property_definition.definition property_definition {property_definition.description = 'product category values'} property_definition <= property_definition_representation.definition

Table 70 – Mapping table for material_product UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.used_ representation -> representation representation => parameter_assignment_representation
ee_product_category to ee_document (as reference_document)	PATH			characterized_product_category <= characterized_object document_assigned_item = characterized_object document_assigned_item document_assigned_item <= applied_document_reference.items[i] applied_document_reference applied_document_reference <= document_reference {document_reference.role -> object_role object_role.name = 'product category reference document'} document_reference.assigned_document -> (document) (document => ee_specification)
EE_PRODUCT_CATEGORY_- RELATIONSHIP	product_category_relationship	41		product_category_relationship
ee_product_category_relationship to ee_product_category (as category)	PATH			product_category_relationship product_category_relationship.category -> product_category => characterized_product_category
ee_product_category_relationship to ee_product_category (as sub_category)	PATH			product_category_relationship product_category_relationship.sub_category -> product_category => characterized_product_category

Table 70 – Mapping table for material_product UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT_RELATED_- PRODUCT_CATEGORY	product_related_characterized_- product_category	210		product_related_characterized_product_category <= characterized_product_category <= product_category {product_category => product_related_product_category}
ee_product_related_product_- category to ee_product (as of_product)	PATH			product_related_characterized_product_category <= product_related_product_category product_related_product_category.products[i] -> product
MATERIAL_COMPOSITION_- RELATIONSHIP	product_material_composition_- relationship	45	207	
composition_basis	product_material_composition_- relationship.composition_basis	45		
composition_class	product_material_composition_- relationship.class	45		
determination_method	product_material_composition_- relationship.determination_method	45		
material_composition_relationship to data_element (as constituent_amount)	PATH			product_material_composition_relationship product_material_composition_- relationship.constituent_amount[i] -> measure_with_unit
material_composition_relationship to ee_material (as related_product)	PATH		339	product_material_composition_relationship <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition characterized_product_definition = product_- definition

Table 70 – Mapping table for material_product UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation
material_composition_relationship to ee_material (as relating_product)	PATH		339	product_material_composition_relationship <= product_definition.relationship product_definition.relationship.relatin_ product_definition -> product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation
MATERIAL_DEFINITION	product_definition	41		product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'material definition'}
material_definition to ee_material (as material)	PATH			product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation

Table 71 – Mapping table for non_topological_surface UoF

Application element	AIM element	Source	Rules	Reference path
GEOMETRICALLY_- BOUNDED_SURFACE	geometrically_bounded_surface_- shape_representation	507		geometrically_bounded_surface_shape_- representation <= shape_representation

Table 72 – Mapping table for package UoF

Application element	AIM element	Source	Rules	Reference path
ALTERED_PACKAGE	(package) (externally_defined_package) (library_defined_package)	210 210 210		(package <= physical_unit <= product_definition) (externally_defined_package <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.description = 'altered package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}]}
altered_package to package (as package_to_be_altered)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <=

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'package alteration'} product_definition_relationship product_definition_relationship.relatin_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
altered_package to external_definition (as of_geometric_status)	PATH			(package <= (externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'externally defined geometric status'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition
altered_package to length_data_element (as modified_terminal_separation)	PATH			(package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'modified terminal separation'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> </pre>
altered_package to predefined_geometric_status (as of_geometric_status)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation representation.name = 'predefined geometric status'} representation.items[i] -> representation_item {(representation_item.name = 'standard datum condition') (representation_item.name = 'undefined') (representation_item.name = 'perpendicular primary datum axis') (representation_item.name = 'perpendicular primary datum plane')}} representation_item => descriptive_representation_item </pre>
ALTERED_PACKAGE_TERMINAL	altered_package_terminal	210		<pre> altered_package_terminal <= package_terminal <= shape_aspect </pre>
altered_package_terminal to package_terminal (as terminal_to_be_altered)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal to be altered'} shape_aspect_relationship shape_aspect_relationship.relying_shape_- aspect -> shape_aspect => package_terminal </pre>
altered_package_terminal to altered_package (as associated_definition)	PATH			<pre> altered_package_terminal <= package_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition [product_definition.description = 'altered package']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'[]}] product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) </pre>
CONNECTION_ZONE_INTER- FACE_PLANE_RELATIONSHIP	connection_zone_interface_- plane_relationship	210		<pre> {[connection_zone_interface_plane_relationship <= shape_aspect_relationship] [connection_zone_interface_plane_relationship <= shape_aspect]} </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
derived_zone_shape_class	shape_aspect_- relationship.description			connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.description {(shape_aspect_relationship.description = 'area') (shape_aspect_relationship.description = 'edge')}}
connection_zone_interface_- plane_relationship to connection_zone (as derived_zone)	PATH			connection_zone_interface_plane_relationship <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} connection_zone_interface_plane_relationship
connection_zone_interface_- plane_relationship to connection_zone (as mating_zone)	PATH			<= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
connection_zone_interface_- plane_relationship to seating_plane (as interface_plane)	PATH			connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => seating_plane
FIDUCIAL_PART_FEATURE	fiducial_part_feature	210		fiducial_part_feature <= part_tooling_feature <= shape_aspect
GUIDED_WAVE_TERMINAL	guided_wave_terminal	210		guided_wave_terminal <= package_terminal <= shape_aspect
LENGTH_TRIMMED_TERMINAL	length_trimmed_terminal	210	333	length_trimmed_terminal <= altered_package_terminal <= package_terminal <= shape_aspect
length_trimmed_terminal to length_data_element (as maximum_trimmed_length)	PATH		333	length_trimmed_terminal <= altered_package_terminal <= package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum trimmed length'}

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit length_trimmed_terminal <= altered_package_terminal <= package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum trimmed length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
length_trimmed_terminal to length_data_element (as minimum_trimmed_length)	PATH			
PACKAGE	(package) (externally_defined_package) (library_defined_package)	210 210 210	338, 223	<pre> (package <= physical_unit <= product_definition) (externally_defined_package <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition)) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}] (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition </pre>
mounting_technology	descriptive_representation_- item.description	45		

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'package mounting data'} representation representation.items[i] -> representation_item => {representation_item representation_item.name = 'mounting technology'} descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'surface mount') (descriptive_representation_item.description = 'through hole')}} descriptive_representation_item.description (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <=)
package to cartesian_point (as least_material_condition_- centroid_location)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'material property data'} representation representation.items[i] -> representation_item => {cartesian_point cartesian_point.name = 'least material condition centroid location'} cartesian_point (package <= externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <=
package to cartesian_point (as maximum_material_ condition_centroid_location)	PATH			(package <= externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <=

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'material property data'} representation representation.items[i] -> representation_item => {cartesian_point cartesian_point.name = 'maximum material condition centroid location'} cartesian_point (package <= externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=)
package to datum_reference_frame (as of_datum_reference_frame)	PATH			(package <= externally_defined_package <= externally_defined_physical_unit <= (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=)

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect <= {shape_aspect => package_terminal => primary_reference_terminal} shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'of datum reference frame'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => datum_reference_frame (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition
package to datum_feature_usage_ in_datum_system_relationship (as of_defining_datum_system_ relationship)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect <= {(shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => physical_unit_datum_feature => tertiary_orientation_feature)}} shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'datum feature usage in datum system'} (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition
package to datum_usage_in_- datum_system_relationship (as of_derived_datum_system_- defining_relationship)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition => product_definition_shape <= shape_aspect.of_shape shape_aspect <= {(shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => physical_unit_datum_feature => tertiary_orientation_feature)}} shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'datum usage in datum system'} (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_- definition
package to external_definition (as surface_mount_lead_form)	PATH			package to external_definition (as surface_mount_lead_form)

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'externally defined surface mount lead form'} representation_item => externally_defined_representation_item => external_definition
package to predefined_lead_form (as surface_mount_lead_form)	PATH			(package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <-

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item => {representation_item representation_item.name = 'predefined surface mount lead form'} descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'gull wing') (descriptive_representation_item.description = 'j lead') (descriptive_representation_item.description = 'reversed j lead') (descriptive_representation_item.description = 'integral terminal') (descriptive_representation_item.description = 'ball') (descriptive_representation_item.description = 'undefined') (descriptive_representation_item.description = 'straight')}} descriptive_representation_item.description (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= </pre>
package to length_data_element (as maximum_body_clearance_- above_seating_plane)	PATH			<pre> (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum body clearance above seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as maximum_body_clearance_ below_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum body clearance below seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as maximum_body_height_ above_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum body height above seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as maximum_body_height_ below_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum body height below seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as maximum_lead_length_ below_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum lead length below seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=)
package to length_data_element (as maximum_seating_plane_ installation_offset)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'maximum seating plane installation offset'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <= (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as minimum_body_plane_ above_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'minimum body clearance above seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as minimum_body_clearance_ below_seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'minimum body clearance below seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <=
package to length_data_element (as least_lead_length_below_ seating_plane)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'least lead length below seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <=) (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=)
package to length_data_element (as nominal_mounting_lead_pitch)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'nominal mounting lead pitch'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit (package <= (externally_defined_package <= externally_defined_physical_unit <=) (library_defined_package <= library_defined_physical_unit <= externally_defined_physical_unit <=)
package to length_data_element (as nominal_mounting_lead_span)	PATH			

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'package mounting data'} representation.items[i] -> representation_item {representation_item.name = 'nominal mounting lead span'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
PACKAGE_BODY	package_body	210	333	<pre> package_body <= shape_aspect </pre>
package_body to ee_material (as body_material)	PATH		333	<pre> package_body <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition =shape_definition characterized_definition <- material_designation.definitions[i] material_designation </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_body to package (as associated_definition)	PATH		340, 338	<pre> package_body <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) </pre>
PACKAGE_BODY_BOTTOM_ SURFACE	package_body_bottom_surface	210		<pre> package_body_bottom_surface <= package_body_surface <= shape_aspect </pre>
PACKAGE_BODY_EDGE_SEG- MENT_SURFACE	package_body_edge_segment_ surface	210		<pre> package_body_edge_segment_surface <= [shape_aspect_relationship] [shape_aspect] </pre>
package_body_edge_segment_ surface to package_body_edge_surface (as composed_surface)	PATH			<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				package_body_surface => package_body_edge_surface
package_body_edge_segment_- surface to edge_segment_vertex (as end_vertex)	PATH			package_body_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
package_body_edge_segment_- surface to edge_segment_vertex (as start_vertex)	PATH			package_body_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
PACKAGE_BODY_EDGE_SUR- FACE	package_body_edge_surface	210		package_body_edge_surface <= package_body_surface <= shape_aspect
PACKAGE_BODY_SURFACE	package_body_surface	210		package_body_surface <= shape_aspect
package_body_surface to package_body (as associated_package_body)	PATH			package_body_surface <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated package body'}

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => package_body</code>
PACKAGE_BODY_TOP_SURFACE_FACE	package_body_top_surface	210		<code>package_body_top_surface <= package_body_surface <= shape_aspect</code>
PACKAGE_TERMINAL	package_terminal	210	333	<code>package_terminal <= shape_aspect</code>
seating_plane_intersection	PATH			<code>shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'seating plane intersection'} representation_item => descriptive_representation_item {(descriptive_representation_item.description = 'surface_intersection') (descriptive_representation_item.description = 'through_intersection') (descriptive_representation_item.description = 'does_not_intersect')}</code>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_terminal to analytical_representation (as model_representation)	PATH		333	<pre> package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> {representation => analytical_model} representation <- representation_relationship.rep_1 representation_relationship.rep_2 -> representation => analytical_representation </pre>
package_terminal to characteristic (as terminal_characteristic)	PATH		109	<pre> package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation.items[i] -> representation_item </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type} </pre>
package_terminal to connection_zone (as external_connection_zone)	PATH		333	<pre> package_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'external connection zone'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>
package_terminal to connection_zone (as internal_connection_zone)	PATH			<pre> package_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'internal connection zone'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_terminal to connection_- zone_interface_plane_relationship (as seating_plane_zone)	PATH			<pre> package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'seating plane zone'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => connection_zone_interface_plane_relationship </pre>
package_terminal to length_data_element (as maximum_terminal_- diametrical_extent)	PATH			<pre> package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum terminal diametrical extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
package_terminal to length_data_element (as minimum_terminal_- diametrical_extent)	PATH			<pre> package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum terminal diametrical extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
package_terminal to package (as associated_definition)	PATH		333, 340, 338	<pre> package_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) </pre>
package_terminal to material_definition (as terminal_core_material)	PATH			<pre> package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal core material'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'material definition'} </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_terminal to material_definition (as terminal_surface_material)	PATH			<pre> package_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal surface material'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'material definition'} </pre>
PACKAGE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP	shape_aspect_relationship	41		<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'package terminal surface constituent relationship'} </pre>
package_terminal_surface_-constituent_relationship to package_terminal (as related_shape_element)	PATH			<pre> shape_aspect_relationship -> shape_aspect => package_terminal </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
package_terminal_surface_- constituent_relationship to package_body_surface (as relating_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect => package_body_surface
PART_TOOLING_FEATURE	part_tooling_feature	210		part_tooling_feature <= shape_aspect
POLARITY_INDICATION_FEATURE	shape_aspect	41	251	shape_aspect {shape_aspect.description = 'polarity indication feature'}
polarity_indication_feature to package_body_surface (as associated_body_vertical_extent)	PATH			shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface)
polarity_indication_feature to package_terminal (as associated_terminal)	PATH			shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated terminal'} shape_aspect_relationship shape_aspect_relationship.relatiing_shape_- aspect -> shape_aspect => package_terminal
PRIMARY_ORIENTATION_FEATURE	primary_orientation_feature	210		primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
primary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			<pre> primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface) </pre>
PRIMARY_REFERENCE_TERMINAL	primary_reference_terminal	210		<pre> primary_reference_terminal <= package_terminal <= shape_aspect </pre>
SEATING_PLANE	seating_plane	210		<pre> seating_plane <= shape_aspect {[shape_aspect shape_aspect.product_definitional = .FALSE.]} </pre>
seating_plane to package (as scope)	PATH			<pre> shape_aspect shape_aspect ({shape_aspect (shape_aspect.name= 'package seating plane') (shape_aspect.name= 'interface plane')}) shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit => (externally_defined_physical_unit => (externally_defined_package) (library_defined_physical_unit => (library_defined_package))) (package)) ({shape_aspect (shape_aspect.name= 'die seating plane')}) shape_aspect.of_shape -> product_definition_shape <= property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit => (externally_defined_physical_unit => (externally_defined_bare_die) (library_defined_physical_unit => (library_defined_bare_die))) (bare_die)) </pre>
SECONDARY_ORIENTATION_- FEATURE	secondary_orientation_feature	210		<pre> secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect </pre>
secondary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			<pre> secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface)</code>
SHAPE_FORMED_TERMINAL	shape_formed_terminal	210	333	<code>shape_formed_terminal <= altered_package_terminal <= package_terminal <= shape_aspect</code>
shape_formed_terminal to lead_form_specification (as terminal_form_specification)	PATH			<code>shape_formed_terminal <= altered_package_terminal <= package_terminal document_assigned_item = package_terminal document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {[document document.kind -> document_type document_type.product_data_type = 'lead form specification'] [document <- document_relationship.related_document document_relationship document_relationship.relatng_document -> document document.kind -> document_type document_type.product_data_type = 'material specification']}] document => ee_specification</code>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
SURFACE_PREPPEPED_TERMINAL	surface_prepped_terminal	210	333	surface_prepped_terminal <= altered_package_terminal <= package_terminal <= shape_aspect
surface_prepped_terminal to surface_finish_specification (as surface_preparation)	PATH			surface_prepped_terminal <= altered_package_terminal <= package_terminal document_assigned_item = package_terminal document_assigned_item <= applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'surface finish specification'} document => ee_specification
TERTIARY_ORIENTATION_- FEATURE	tertiary_orientation_feature	210		tertiary_orientation_feature <= physical_unit_datum_feature <= shape_aspect
tertiary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			tertiary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface) </pre>
TEST_POINT_PART_FEATURE	test_point_part_feature	210		<pre> test_point_part_feature <= part_tooling_feature <= shape_aspect </pre>
WIRE_TERMINAL	wire_terminal	210		<pre> wire_terminal <= package_terminal <= shape_aspect </pre>
wire_terminal to connection_zone (as internal_connection_zone)	PATH			<pre> wire_terminal <= package_terminal <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'internal connection zone'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'} </pre>
wire_terminal to length_data_element (as maximum_wire_terminal_length)	PATH			<pre> wire_terminal <= package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition </pre>

Table 72 – Mapping table for package UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum wire terminal length'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit</pre>
wire_terminal to length_data_element (as minimum_wire_terminal_length)	PATH			<pre>wire_terminal <= package_terminal <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'minimum wire terminal length'}</pre>

Table 72 – Mapping table for package UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit

Table 73 – Mapping table for packaged_part_black_box_model UoF

Application element	AIM element	Source	Rules	Reference path
ALTERED_PACKAGED_PART	(packaged_part) (externally_defined_packaged_part) (library_defined_packaged_part)	210 210 210		(packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.description = 'altered packaged part'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}]}
altered_packaged_part to packaged_part (as base_packaged_part)	PATH			(packaged_part <= (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <=)

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'base packaged part'} product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (packaged_part) (externally_defined_physical_unit => externally_defined_packaged_part) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part) (packaged_part <= (externally_defined_packaged_part <= externally_defined_physical_unit <= (library_defined_packaged_part <= library_defined_physical_unit <=
altered_packaged_part to altered_package (as used_package)	PATH			(packaged_part <= (externally_defined_packaged_part <= externally_defined_physical_unit <= (library_defined_packaged_part <= library_defined_physical_unit <=

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'used package'} product_definition_relationship.related_ product_definition -> {product_definition [product_definition.description = 'altered package'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage']} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package)
PACKAGED_CONNECTOR	(packaged_connector) (externally_defined_packaged_ connector) (library_defined_packaged_ connector)	210 210 210	338	(packaged_connector <= packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_connector <=

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_connector <= library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition])
PACKAGED_CONNECTOR_- TERMINAL_RELATIONSHIP	packaged_connector_terminal_- relationship	210	224	packaged_connector_terminal_relationship <= [shape_aspect] [shape_aspect_relationship]
packaged_connector_terminal_- relationship to packaged_connector (as connector)	PATH		333, 340, 338	packaged_connector_terminal_relationship <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit => (packaged_part => packaged_connector) (externally_defined_physical_unit => externally_defined_packaged_part =>

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_packaged_connector) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector)
packaged_connector_terminal_- relationship to packaged_part_interface_terminal (as interface_terminal)	PATH		344, 333	packaged_connector_terminal_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => packaged_part_terminal
packaged_connector_terminal_- relationship to packaged_part_join_terminal (as join_terminal)	PATH		344, 333	packaged_connector_terminal_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => packaged_part_terminal
PACKAGED_PART	(packaged_part) (externally_defined_packaged_- part) (library_defined_packaged_part)	210 210 210	338, 225	(packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <=

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition)) {product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}] </pre>
packaged_part to ee_material (as potting_compound)	PATH			<pre> (packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_- designation classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'potting compound'} </pre>

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
packaged_part to functional_unit_usage_view (as implemented_function)	PATH		338	<pre> (packaged_part <=) (externally_defined_packaged_part <=) (externally_defined_physical_unit <=) (library_defined_packaged_part <=) (library_defined_physical_unit <=) (externally_defined_physical_unit <=) physical_unit <= product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'implemented_function'} product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional design_usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) (packaged_part <=) (externally_defined_packaged_part <=) (externally_defined_physical_unit <=) (library_defined_packaged_part <=) </pre>
packaged_part to package (as used_package)	PATH			

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'used package'} product_definition_relationship product_definition_relationship.relatin_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) </pre>
PACKAGED_PART_INTER- FACE_TERMINAL	packaged_part_terminal	210	333	<pre> packaged_part_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interface terminal'} </pre>

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PACKAGED_PART_JOIN_TERMINAL	packaged_part_terminal	210		packaged_part_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'join terminal'}
PACKAGED_PART_TERMINAL	packaged_part_terminal	210	333	packaged_part_terminal <= shape_aspect {(shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')}}
packaged_part_terminal to package_terminal (as terminal_of_package)	PATH		333, 340, 338	packaged_part_terminal <= {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition <= product_definition_relationship.related_ product_definition product_definition_relationship {product_definition_relationship.name = 'used package'}} product_definition_relationship.relatin_ product_definition -> {product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit =>

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> library_defined_package}} product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => package_terminal} shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'terminal of package'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => package_terminal packaged_part_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>
packaged_part_terminal to packaged_part (as associated_definition)	PATH		338	

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (packaged_part) (externally_defined_physical_unit => externally_defined_packaged_part) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part)
PART_CONNECTED_TERMINALS_DEFINITION	part_connected_terminals_ definition	210		part_connected_terminals_definition <= shape_aspect
name	shape_aspect.name	41	333	part_connected_terminals_definition <= shape_aspect shape_aspect.name
part_connected_terminals_ definition to part_terminal (as connected_terminals)	PATH		333	part_connected_terminals_definition <= shape_aspect <- shape_aspect.relationship.relatng_shape_ aspect shape_aspect.relationship {shape_aspect_relationship shape_aspect_relationship.name = 'member connected_terminal'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal)

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(minimally_defined_bare_die_terminal) (interconnect_module_terminal) (packaged_part_terminal)
part_connected_terminals_- definition to Assembly_module_usage_view (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition
part_connected_terminals_- definition to Bare_die (as associated_definition)	PATH			

Table 73 – Mapping table for packaged_part_black_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => (physical_unit => bare_die) (physical_unit => externally_defined_physical_unit => externally_defined_bare_die) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die) </pre>
<pre> part_connected_terminals_- definition to Interconnect_module_usage_view (as associated_definition) </pre>	PATH			<pre> part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= </pre>

Table 73 – Mapping table for packaged_part_black_box_model UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'physical design usage'[]} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => (physical_unit => packaged_part) (physical_unit => externally_defined_physical_unit => externally_defined_packaged_part) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part) </pre>
part_connected_terminals_- definition to Packaged_part (as associated_definition)	PATH			

Table 74 – Mapping table for packaged_part_terminal_analytical_representation UoF

Application element	AIM element	Source	Rules	Reference path
PART_DEVICE_TERMINAL_- MODEL_PORT_ASSIGNMENT	property_definition_representation	41	229	
part_device_terminal_model_- port_assignment to analytical_model_port (as internal_port)	PATH		342	property_definition_representation property_definition_representation.used_- representation -> representation => analytical_model_port
part_device_terminal_model_- port_assignment to part_device_terminal (as internal_connection)	PATH			property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect
PART_FEATURE_MODEL_- PORT_ASSIGNMENT	property_definition_representation	41		{property_definition_representation property_definition_representation.name = 'part feature model port assignment'}
part_feature_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation {property_definition_representation.name = 'part feature model port assignment'} property_definition_representation.used_- representation -> representation => analytical_model_port

Table 74 – Mapping table for packaged_part_terminal_analytical_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_feature_model_port_- assignment to part_feature (as source)	PATH			<pre> property_definition_representation { property_definition_representation.name = 'part feature model port assignment' } property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect { [[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']] (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') </pre>

Table 74 – Mapping table for packaged_part_terminal_analytical_representation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}

Table 75 – Mapping table for packaged_part_white_box_model UoF

Application element	AIM element	Source	Rules	Reference path
CONNECTION_ZONE_INTERFACE_PLANE_RELATIONSHIP	connection_zone_interface_- plane_relationship	210		{[connection_zone_interface_plane_relationship <= shape_aspect_relationship] [connection_zone_interface_plane_relationship <= shape_aspect]}
derived_zone_shape_class	shape_aspect_- relationship.description			connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.description {(shape_aspect_relationship.description = 'area') (shape_aspect_relationship.description = 'edge')}
connection_zone_interface_- plane_relationship to connection_zone (as derived_zone)	PATH			connection_zone_interface_plane_relationship <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'derived zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
connection_zone_interface_- plane_relationship to connection_zone (as mating_zone)	PATH			connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
connection_zone_interface_- plane_relationship to seating_plane (as interface_plane)	PATH			connection_zone_interface_plane_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => seating_plane

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
CONNECTION_ZONE_- MODEL_PORT_ASSIGNMENT	property_definition_representation	41		property_definition_representation
connection_zone_model_port_- assignment to connection_zone (as source)	PATH			property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.description = 'connection zone'}
connection_zone_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port
DEVICE_2D_POSITION	mapped_item	43	128	
device_2d_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d
device_2d_position to part_device (as placed_device)	PATH			mapped_item <= representation_item <= representation.items[i] {representation => shape_representation}

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship </pre>
device_2d_position to physical_unit_planar_shape (as device_shape)	PATH		338, 111	<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => physical_unit] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition => physical_unit => bare_die}] representation => shape_representation mapped_item mapped_item.mapping_target -> representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> </pre>
device_2d_position to physical_unit_planar_shape (as part_shape)	PATH		338	

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit => packaged_part[]} representation => shape_representation
DEVICE_3D_POSITION	mapped_item	43	129	
device_3d_position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
device_3d_position to part_device (as placed_device)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition_relationship product_definition_relationship </pre>
device_3d_position to physical_unit_3d_shape (as device_shape)	PATH		338	<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {[representation representation.name = '3d bound volume shape']} [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit => bare_die[]} representation => shape_representation mapped_item mapped_item.mapping_target -> representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape']}
device_3d_position to physical_unit_3d_shape (as part_shape)	PATH		338	

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit => packaged_part}} representation => shape_representation </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DEVICE_TERMINAL_MAP	device_terminal_map	210		device_terminal_map <=[shape_aspect_relationship][shape_aspect]
device_terminal_map to packaged_part_terminal (as target_terminal)	PATH			device_terminal_map <=[shape_aspect_relationship shape_aspect_relationship.relatng_shape_-aspect -> shape_aspect {(shape_aspect.description = 'interface terminal')} (shape_aspect.description = 'join terminal')}] shape_aspect => packaged_part_terminal
device_terminal_map to part_device_terminal (as mapped_device_terminal)	PATH		344, 333	device_terminal_map <=[shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect
DEVICE_TERMINAL_MAP - RELATIONSHIP	shape_aspect_relationship	41	130, 344	{shape_aspect_relationship shape_aspect_relationship.name = 'device terminal_map relationship'}
device_terminal_map_relationship to device_terminal_map (as related_device_terminal_map)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => device_terminal_map
device_terminal_map_relationship to device_terminal_map (as relating_device_terminal_map)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_-aspect -> shape_aspect => device_terminal_map
PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP	packaged_connector_terminal_-relationship	210	224	packaged_connector_terminal_relationship <=[shape_aspect][shape_aspect_relationship]
packaged_connector_terminal_-relationship to packaged_connector (as connector)	PATH		333, 340, 338	packaged_connector_terminal_relationship <=[shape_aspect shape_aspect.of_shape -> product_definition_shape <=

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit => (packaged_part => packaged_connector) (externally_defined_physical_unit => externally_defined_packaged_part => externally_defined_packaged_connector) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part => library_defined_packaged_connector)
packaged_connector_terminal_ relationship to packaged_part_interface_terminal (as interface_terminal)	PATH		344, 333	packaged_connector_terminal_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => packaged_part_terminal

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
packaged_connector_terminal_- relationship to packaged_part_join_terminal (as join_terminal)	PATH		344, 333	packaged_connector_terminal_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => packaged_part_terminal
PACKAGED_PART	(packaged_part) (externally_defined_packaged_- part) (library_defined_packaged_part)	210 210 210	338, 225	(packaged_part <= physical_unit <= product_definition) (externally_defined_packaged_part <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']}]
packaged_part to ee_material (as potting_compound)	PATH			(packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <=)

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_- designation classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'potting compound'} (packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition <- product_definition_relationship.related_- product_definition
packaged_part to functional_unit_usage_view (as implemented_function)	PATH		338	

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'implemented_function'} product_definition_relationship.relatin_ - product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element design usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) library_defined_functional_unit) </pre>
packaged_part to package (as used_package)	PATH			<pre> (packaged_part <=) (externally_defined_packaged_part <= externally_defined_physical_unit <=) (library_defined_packaged_part <= library_defined_physical_unit <= externally_defined_physical_unit <=) physical_unit <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'used package'} product_definition_relationship product_definition_relationship.relatin_ - product_definition -> </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (package) (externally_defined_physical_unit => externally_defined_package) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_package) </pre>
PACKAGED_PART_INTER- FACE_TERMINAL	packaged_part_terminal	210	333	<pre> packaged_part_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interface terminal'} </pre>
PACKAGED_PART_JOIN_TER- MINAL	packaged_part_terminal	210		<pre> packaged_part_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'join terminal'} </pre>
PART_CONNECTED_TERMI- NALS_DEFINITION	part_connected_terminals_- definition	210		<pre> part_connected_terminals_definition <= shape_aspect </pre>
name	shape_aspect.name	41	333	<pre> part_connected_terminals_definition <= shape_aspect shape_aspect.name </pre>
part_connected_terminals_- definition to part_terminal (as connected_terminals)	PATH		333	<pre> part_connected_terminals_definition <= shape_aspect <- shape_aspect.relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'member connected terminal'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => (assembly_module_terminal) (minimally_defined_bare_die_terminal) (interconnect_module_terminal) (packaged_part_terminal) </pre>
<pre> part_connected_terminals_- definition to Assembly_module_usage_view (as associated_definition) </pre>	PATH			<pre> part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'assembly module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_connected_terminals_- definition to Bare_die (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => (physical_unit => bare_die) (physical_unit => externally_defined_physical_unit => externally_defined_bare_die) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)
part_connected_terminals_- definition to Interconnect_module_usage_view (as associated_definition)	PATH			part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition product_definition characterized_product_definition characterized_product_definition = product_- definition

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']] product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) part_connected_terminals_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => (physical_unit => packaged_part) (physical_unit => externally_defined_physical_unit => externally_defined_packaged_part) (physical_unit => </pre>
part_connected_terminals_- definition to Packaged_part (as associated_definition)	PATH			

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part)
PART_DEVICE	product_definition_relationship	41	227, 231	{product_definition_relationship product_definition_relationship.description = 'part device'}
item_identification	product_definition - relationship.name	41		
part_device to bare_die (as included_device)	PATH		338	product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (bare_die) (externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)
part_device to ee_material (as bonding_agent)	PATH			product_definition_relationship characterized_product_definition = product_ definition_relationship characterized_product_definition product_definition

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- material_designation.definitions[i] material_designation {material_designation classification_assigned_item = material_- designation classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group => material_functional_category} group.name = 'bonding agent'} </pre>
part_device to packaged_part (as defined packaged_part)	PATH		338	<pre> product_definition_relationship product_definition_relationship.relatiing_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => physical_unit => (packaged_part) (externally_defined_physical_unit => externally_defined_packaged_part) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_packaged_part) </pre>
PART_DEVICE_TERMINAL	shape_aspect	41	228, 230	

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_device_terminal to bare_die_terminal (as used_bare_die_terminal)	PATH			<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition => physical_unit => (bare_die) (externally_defined_physical_unit => externally_defined_bare_die) (externally_defined_physical_unit => library_defined_physical_unit => library_defined_bare_die)} product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_device_terminal to part_device (as terminated_device)	PATH			<pre> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship {product_definition_relationship product_definition_relationship.description = 'part device'} product_definition_relationship </pre>
PART_DEVICE_TERMINAL_- MODEL_PORT_ASSIGNMENT	property_definition_representation	41	229	
part_device_terminal_model_- port_assignment to analytical_model_port (as internal_port)	PATH		342	<pre> property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition </pre>
part_device_terminal_model_- port_assignment to part_device_terminal (as internal_connection)	PATH			

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition = shape_aspect shape_aspect
PART_FEATURE_MODEL_- PORT_ASSIGNMENT	property_definition_representation	41		{property_definition_representation property_definition_representation.name = 'part feature model port assignment'}
part_feature_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation {property_definition_representation.name = 'part feature model port assignment'}
part_feature_model_port_- assignment to part_feature (as source)	PATH			property_definition_representation.used_- representation -> representation => analytical_model_port property_definition_representation {property_definition_representation.name = 'part feature model port assignment'}
				property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'] (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface')}

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal =>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface))
USAGE_VIEW_CONNECTION_ZONE_TERMINAL_SHAPE_RELATIONSHIP	usage_view_connection_zone_- terminal_shape_relationship	210		<pre> [[usage_view_connection_zone_terminal_shape_- relationships <= representation_relationship_with_- transformation] [usage_view_connection_zone_terminal_shape_- relationships <= representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'usage view connection zone terminal shape relationship'] [usage_view_connection_zone_terminal_shape_- relationships <= representation]] </pre>
usage_view_connection_zone_- terminal_shape_relationship to shape_definition (as associating_terminal_shape)	PATH			<pre> usage_view_connection_zone_terminal_shape_- relationships <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'terminal shape'} representation => shape_representation </pre>

Table 75 – Mapping table for packaged_part_white_box_model UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_view_connection_zone_- terminal_shape_relationship to connection_zone (as associated_usage)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => shape_representation} representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
usage_view_connection_zone_- terminal_shape_relationship to shape_definition (as associated_connection_zone_- shape_definition)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} representation => shape_representation

Table 75 – Mapping table for packaged_part_white_box_model UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
usage_view_connection_zone_- terminal_shape_relationship to axis_placement (as associated_usage_placement)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement

Table 76 – Mapping table for part_feature UoF

Application element	AIM element	Source	Rules	Reference path
EDGE_SEGMENT_VERTEX	edge_segment_vertex	210	146, 147	edge_segment_vertex <= physical_unit_datum <= shape_aspect {shape_aspect shape_aspect.name = 'single datum'}
name	descriptive_representation_- item.description	45		edge_segment_vertex <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.description = 'shape element characterization'} property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'shape element name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
FEATURE_GROUP_RELATIONSHIP	shape_aspect_relationship	41	168	{shape_aspect_relationship shape_aspect_relationship.name = 'feature group relationship'}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
feature_group_relationship to part_feature (as related_feature)	PATH			<pre> shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}}
feature_group_relationship to part_group_feature (as relating_feature)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'part group feature'} shape_aspect => composite_shape_aspect
FIDUCIAL_PART_FEATURE	fiducial_part_feature	210		fiducial_part_feature <= part_tooling_feature <= shape_aspect
INTERCONNECT_MODULE_- CAVITY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module cavity surface'}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_cavity_- surface to interconnect_module_surface_- feature (as reference_surface)	PATH			<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'reference surface'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}} </pre>
INTERCONNECT_MODULE_- CUTOUT_SEGMENT_SUR- FACE	interconnect_module_cutout_- segment_surface	210		<pre> [interconnect_module_cutout_segment_surface <= shape_aspect] [interconnect_module_cutout_segment_surface <= shape_aspect_relationship] {shape_aspect.description = 'interconnect module cutout segment surface'} </pre>
interconnect_module_cutout_- segment_surface to interconnect_module_cutout_- surface (as composed_surface)	PATH			<pre> interconnect_module_cutout_segment_surface <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.relatng_shape_- aspect -> </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{shape_aspect shape_aspect.description = 'interconnect module cutout surface'} shape_aspect interconnect_module_cutout_segment_surface <= shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex</pre>
interconnect_module_cutout_- segment_surface to edge_segment_vertex (as end_vertex)	PATH			
interconnect_module_cutout_- segment_surface to edge_segment_vertex (as start_vertex)	PATH			<pre>interconnect_module_cutout_segment_surface <= shape_aspect.relationship shape_aspect.relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex</pre>
INTERCONNECT_MODULE_- CUTOUT_SURFACE	shape_aspect	41		<pre>shape_aspect {shape_aspect.description = 'interconnect module cutout surface'}</pre>
INTERCONNECT_MODULE_- EDGE_SEGMENT_SURFACE	interconnect_module_edge_- segment_surface	210		<pre>[interconnect_module_edge_segment_surface <= shape_aspect] [interconnect_module_edge_segment_surface <= shape_aspect.relationship] {shape_aspect.description = 'interconnect module edge segment surface'}</pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_edge_- segment_surface to interconnect_module_edge_- surface (as composed_surface)	PATH			interconnect_module_edge_segment_surface <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'composed surface'} shape_aspect_relationship.related_shape_- aspect -> {shape_aspect shape_aspect.description = 'interconnect module edge surface'} shape_aspect
interconnect_module_edge_- segment_surface to edge_segment_vertex (as end_vertex)	PATH			interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex
interconnect_module_edge_- segment_surface to edge_segment_vertex (as start_vertex)	PATH			interconnect_module_edge_segment_surface <= shape_aspect_relationship shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit_datum => edge_segment_vertex
INTERCONNECT_MODULE_- EDGE_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module edge surface'}
INTERCONNECT_MODULE_- PRIMARY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module primary surface'}
INTERCONNECT_MODULE_- SECONDARY_SURFACE	shape_aspect	41		shape_aspect {shape_aspect.description = 'interconnect module secondary surface'}
INTERCONNECT_MODULE_- SURFACE_FEATURE	shape_aspect	41		shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}}}
interconnect_module_surface_- feature to interconnect_module_usage_view (as associated_definition)	PATH		194	shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module']}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']]}
				product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit)
INTERCONNECT_MODULE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'interconnect module terminal surface constituent relationship'}
interconnect_module_terminal_surface_constituent_relationship to interconnect_module_terminal (as related_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => interconnect_module_terminal
interconnect_module_terminal_surface_constituent_relationship to interconnect_module_surface_feature (as relating_shape_element)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface')}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}
PART_ARRAY_FEATURE	composite_array_shape_aspect	210		composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect
PART_FEATURE	(shape_aspect) (composite_shape_aspect) (primary_orientation_feature) (secondary_orientation_feature) (package_body) (part_tooling_feature) (thermal_feature) (part_mounting_feature) (package_terminal) (assembly_module_terminal) (bare_die_terminal) (interconnect_module_terminal) (minimally_defined_bare_die_-_terminal) (packaged_part_terminal) (package_body_surface)	41 210 210 210 210 210 210 210 210 210 210 210 210	232, 233	(composite_shape_aspect <= shape_aspect shape_aspect.description = 'part group feature') (shape_aspect (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect) (secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect) (package_body <= shape_aspect) (part_tooling_feature <=

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect (thermal_feature <= shape_aspect) (part_mounting_feature <= shape_aspect) (package_terminal <= shape_aspect) (assembly_module_terminal <= shape_aspect) (interconnect_module_terminal <= shape_aspect) (bare_die_terminal <= minimally_defined_bare_die_terminal <= shape_aspect) (minimally_defined_bare_die_terminal <= shape_aspect) (packaged_part_terminal <= shape_aspect) (package_body_surface <= shape_aspect)
material_state_change	PATH			shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface')

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {[representation representation.name = 'material state change'] [representation representation.items[i] -> representation_item {(representation_item.name = 'material addition') (representation_item.name = 'material removal')}]}} shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'] (shape_aspect.description = 'part generic feature')
name	PATH	41		

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_aspect.name
part_feature to part_feature (as precedent_feature)	PATH			shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal {!{package_terminal => primary_reference_terminal}}) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_aspect <- </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'precedent feature'}} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)} shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'])
part_feature to physical_unit_usage_view (as associated_definition)	PATH		340, 338	

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition {{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit)

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(physical_unit => externally_defined_physical_unit => library_defined_physical_unit)}
PART_FEATURE_PLACEMENT_LINK	composite_array_shape_aspect_link	210		{[composite_array_shape_aspect_link <= shape_aspect] [composite_array_shape_aspect_link <= shape_aspect_relationship]}
part_feature_placement_link to part_feature (as precedent_element)	PATH			composite_array_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect
part_feature_placement_link to part_feature (as subsequent_element)	PATH			composite_array_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect
PART_GROUP_FEATURE	composite_shape_aspect	47		{composite_shape_aspect <= shape_aspect shape_aspect.description = 'part group feature'}
PART_INTERFACE_ACCESS_FEATURE	part_interface_access_feature	210		part_interface_access_feature <= shape_aspect
part_interface_access_feature to connection_zone (as interface_access_area)	PATH			part_interface_access_feature <= shape_aspect <- shape_aspect_relationship.relatng_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'interface_access_area'}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'connection zone'} shape_aspect
part_interface_access_feature to part_feature (as accessed_interface)	PATH			part_interface_access_feature <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'accessed interface'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']} (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface')

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}
PART_LINEAR_ARRAY_FEA-TURE	linear_composite_array_shape_- aspect	210		linear_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect
part_linear_array_feature to part_feature_placement_link (as element)	PATH			linear_composite_array_shape_aspect <= composite_array_shape_aspect <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => composite_array_shape_aspect_link linear_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'linear array composite properties'} property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'linear array properties']} [representation representation.context_of_items -> representation_context [representation_context representation_context => global_unit_assigned_context] [representation_context representation_context => </pre>
part_linear_array_feature to length_data_element (as initial_node_positive_x_- displacement)	PATH			

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>global_uncertainty_assigned_context</code> <code>[representation_context</code> <code> representation_context =></code> <code> parametric_representation_context]]</code> <code>[representation</code> <code> representation_context_of_items -></code> <code> representation_context</code> <code> representation_context.context_type = 'array</code> <code> parametric_context']]</code> <code> representation.items[i] -></code> <code> representation_item =></code> <code> {representation_item</code> <code> representation_item.name = 'initial element</code> <code> positive x displacement'}</code> <code> measure_representation_item <=</code> <code> measure_with_unit</code> <code> {measure_with_unit =></code> <code> length_measure_with_unit}</code> <code> measure_with_unit.value_component</code>
part_linear_array_feature to length_data_element (as location_uncertainty)	PATH			<code>linear_composite_array_shape_aspect <=</code> <code>composite_array_shape_aspect <=</code> <code>composite_shape_aspect <=</code> <code>shape_aspect</code> <code>shape_definition = shape_aspect</code> <code>characterized_definition = shape_definition</code> <code>characterized_definition <-</code> <code>property_definition.definition</code> <code>property_definition</code> <code>{property_definition</code> <code> property_definition.name = 'linear array</code> <code> composite_properties'}</code> <code>property_definition <-</code> <code>property_definition_representation.definition</code>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation -> representation {representation representation.name = 'linear array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_ context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit </pre>
<pre> part_linear_array_feature to length_data_element (as terminus_node_positive_x_- displacement) </pre>	PATH			<pre> linear_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'linear array composite properties'} property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'linear array properties']} </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [representation representation.context_of_items -> representation_context [representation_context representation_context => global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type representation_context.context_type = 'array parametric_context']] representation.items[i] -> representation_item => {representation_item representation_item.name = 'terminus element positive x displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component {[linear_composite_array_shape_aspect_link <= shape_aspect] [linear_composite_array_shape_aspect_link <= shape_aspect_relationship]]} </pre>
PART_LINEAR_ARRAY_FEATURE_LINK	linear_composite_array_shape_aspect_link	210		

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_linear_array_feature_link to part_linear_array_feature (as precedent_element)	PATH			linear_composite_array_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => composite_shape_aspect => composite_array_shape_aspect => linear_composite_array_shape_aspect
part_linear_array_feature_link to part_linear_array_feature (as subsequent_element)	PATH			linear_composite_array_shape_aspect_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => composite_shape_aspect => composite_array_shape_aspect => linear_composite_array_shape_aspect
PART_MATING_FEATURE	part_mating_feature	210		part_mating_feature <= shape_aspect
part_mating_feature to connection_zone (as mating_area)	PATH			part_mating_feature <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mating area'} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'connection zone'}
PART_MOUNTING_FEATURE	part_mounting_feature	210		part_mounting_feature <= shape_aspect
part_mounting_feature to connection_zone (as mounting_area)	PATH			part_mounting_feature <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect {shape_aspect_relationship shape_aspect_relationship.name = 'mounting area'}

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'connection zone'} shape_aspect
PART_RECTANGULAR_ARRAY_FEATURE	rectangular_composite_array_shape_aspect	210		rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect
part_rectangular_array_feature to part_linear_array_feature_link (as element)	PATH			rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'element'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => linear_composite_array_shape_aspect_link rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-
part_rectangular_array_feature to length_data_element (as initial_row_positive_y_- displacement)	PATH			rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition {property_definition property_definition.name = 'rectangular array composite properties'} property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context => representation_context => global_unit_assigned_context] [representation_context => global_uncertainty_assigned_context] [representation_context => parametric_representation_context]] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric context']} representation.items[i] -> representation_item => </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'initial row positive y displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value.component </pre>
part_rectangular_array_feature to length_data_element (as location_uncertainty)	PATH			<pre> rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'rectangular array composite properties'} property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation {representation representation.name = 'rectangular array properties'} representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context global_uncertainty_assigned_- context.uncertainty[i] -> </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit
part_rectangular_array_feature to length_data_element (as terminus_row_positive_y_- displacement)	PATH			rectangular_composite_array_shape_aspect <= composite_array_shape_aspect <= composite_shape_aspect <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'rectangular array composite properties'} property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation {[representation representation.name = 'rectangular array properties'] [representation representation.context_of_items -> representation_context [representation_context representation_context => global_unit_assigned_context] [representation_context global_uncertainty_assigned_context] [representation_context representation_context =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> parametric_representation_context[] [representation representation.context_of_items -> representation_context representation_context.context_type = 'array parametric_context'] representation.items[i] -> representation_item => {representation_item representation_item.name = 'terminus row positive y displacement'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit} measure_with_unit.value_component </pre>
PART_TERMINAL	<pre> (assembly_module_terminal) (bare_die_terminal) (interconnect_module_terminal) (minimally_defined_bare_die_ - terminal) (packaged_part_terminal) </pre>	210 210 210 210 210	333	<pre> (assembly_module_terminal <=) (bare_die_terminal <=) (minimally_defined_bare_die_terminal <=) (interconnect_module_terminal <=) (minimally_defined_bare_die_terminal <=) (packaged_part_terminal <=) shape_aspect </pre>
PART_TOOLING_FEATURE	part_tooling_feature	210		<pre> part_tooling_feature <= shape_aspect </pre>
POLARITY_INDICATION_FEATURE	shape_aspect	41	251	<pre> shape_aspect {shape_aspect.description = 'polarity indication feature'} </pre>
polarity_indication_feature to package_body_surface (as associated_body_vertical_extent)	PATH			<pre> shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				package_body_top_surface) (shape_aspect => package_body_bottom_surface)
polarity_indication_feature to package_terminal (as associated_terminal)	PATH			shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated terminal'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => package_terminal
PRIMARY_ORIENTATION_- FEATURE	primary_orientation_feature	210		primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect
primary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			primary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body vertical extent'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface)
PRIMARY_REFERENCE_TER- MINAL	primary_reference_terminal	210		primary_reference_terminal <= package_terminal <= shape_aspect
SECONDARY_ORIENTATION_- FEATURE	secondary_orientation_feature	210		secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
secondary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			secondary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body_vertical_extent'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface)
TERTIARY_ORIENTATION_- FEATURE	tertiary_orientation_feature	210		tertiary_orientation_feature <= physical_unit_datum_feature <= shape_aspect
tertiary_orientation_feature to package_body_surface (as associated_body_vertical_extent)	PATH			tertiary_orientation_feature <= physical_unit_datum_feature <= shape_aspect <= shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'associated body_vertical_extent'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => package_body_top_surface) (shape_aspect => package_body_bottom_surface)
TEST_POINT_PART_FEATURE	test_point_part_feature	210		test_point_part_feature <= part_tooling_feature <= shape_aspect

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
THERMAL_FEATURE	thermal_feature	210		thermal_feature <= shape_aspect
USAGE_CONCEPT_USAGE_- RELATIONSHIP	mapped_item	43		{mapped_item <= representation_item representation_item.name = 'ucur'} mapped_item.mapping_source -> representation_map representation_map.mapping_origin -> representation_item => geometric_representation_item => placement}
usage_concept_usage_relationship to axis_placement (as associated_usage_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement
usage_concept_usage_relationship to cartesian_transformation_ operator_2d (as associated_usage_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => cartesian_transformation_operator_2d
usage_concept_usage_relationship to component_termination_ passage_template_terminal (as associated_usage)	PATH			mapped_item <= representation_item <= representation.items[i] {representation => shape_representation} representation <= property_definition_representation.used_ representation

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal')} mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation </pre>
usage_concept_usage_relationship to shape_definition (as associated_shape_definition)	PATH			
usage_concept_usage_relationship to land_template_terminal (as associated_usage)	PATH		333	<pre> mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition shape_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')} shape_aspect => land_template_terminal
usage_concept_usage_relationship to non_feature_shape_element (as associated_usage)	PATH			mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.} mapped_item <= representation_item <- representation.items[i] representation {representation => shape_representation} representation <-
usage_concept_usage_relationship to non_feature_shape_definition (as associating_non_feature_ usage_shape)	PATH			

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}}</pre>
usage_concept_usage_relationship to part_feature (as associated_usage)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'])}</pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect =>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
usage_concept_usage_relationship to part_template_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
usage_concept_usage_relationship to part_template_non_planar_2d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'part template non planar 2d shape']} [representation <- </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation </pre>
usage_concept_usage_relationship to part_template_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
usage_concept_usage_relationship to physical_unit_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation } </pre>
usage_concept_usage_relationship to physical_unit_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape'] } </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit}] representation => shape_representation </pre>
usage_concept_usage_relationship to printed_part_template_terminal (as associated_usage)	PATH		333	<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect </pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal') shape_aspect => printed_part_template_terminal</pre>
usage_concept_usage_relationship to via_template_terminal (as associated_usage)	PATH			<pre>mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'via template terminal'}</pre>
USAGE_VIEW_CONNECTION_ZONE_TERMINAL_SHAPE_RELATIONSHIP	usage_view_connection_zone_- terminal_shape_relationship	210		<pre>{[usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation] [usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'usage view connection zone terminal shape relationship'] [usage_view_connection_zone_terminal_shape_- relationship <= representation]}</pre>

Table 76 – Mapping table for part_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_view_connection_zone_- terminal_shape_relationship to shape_definition (as associating_terminal_shape)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'terminal shape'} representation => shape_representation
usage_view_connection_zone_- terminal_shape_relationship to connection_zone (as associated_usage)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} {representation => shape_representation} representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}

Table 76 – Mapping table for part_feature UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
usage_view_connection_zone_- terminal_shape_relationship to shape_definition (as associated_connection_zone_- shape_definition)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = 'zone shape'} representation => shape_representation
usage_view_connection_zone_- terminal_shape_relationship to axis_placement (as associated_usage_placement)	PATH			usage_view_connection_zone_terminal_shape_- relationship <= representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined.transformation item_defined.transformation.transformation_item_1 -> representation_item {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement

Table 77 – Mapping table for part_feature_to_analytical_model_port_mapping UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_REPRESENTATION	analytical_representation	210		analytical_representation <= representation
analytical_representation_name	representation.name	43		analytical_representation <= representation representation.name
analytical_representation to analytical_model (as model_representation)	PATH			analytical_representation <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => analytical_model
analytical_representation to parameter_assignment (as model_parameter_assignment)	PATH			analytical_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => parameter_assignment_representation
PART_FEATURE_MODEL_-PORT_ASSIGNMENT	property_definition_representation	41		{property_definition_representation property_definition_representation.name = 'part feature model port assignment'}
part_feature_model_port_-assignment to analytical_model_port (as target)	PATH		342	property_definition_representation {property_definition_representation.name = 'part feature model port assignment'} property_definition_representation.used_-representation -> representation => analytical_model_port

Table 77 – Mapping table for part_feature_to_analytical_model_port_mapping UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_feature_model_port_- assignment to part_feature (as source)	PATH			<pre> property_definition_representation {property_definition_representation.name = 'part feature model port assignment'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']} (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') </pre>

Table 77 – Mapping table for part_feature_to_analytical_model_port_mapping UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}

Table 78 – Mapping table for part_template UoF

Application element	AIM element	Source	Rules	Reference path
CONNECTION_ZONE_MAP_- IDENTIFICATION	connection_zone_map_- identification	210		{connection_zone_map_identification <= [representation_relationship] [shape_aspect]] {connection_zone_map_identification <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition => footprint_definition}
connection_zone_map_- identification to usage_view_connection_zone_- terminal_shape_relationship (as reference_connection_zone_- shape)	PATH			connection_zone_map_identification <= representation_relationship representation_relationship.rep_1 -> representation => usage_view_connection_zone_terminal_shape_- relationship
connection_zone_map_- identification to usage_view_connection_zone_- terminal_shape_relationship (as mapped_connection_zone_shape)	PATH			connection_zone_map_identification <= representation_relationship representation_relationship.rep_2 -> representation => usage_view_connection_zone_terminal_shape_- relationship

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
DEFAULT_TRACE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'default trace template'}
default_trace_template to stratum_technology (as of_stratum_technology)	PATH		333, 340	part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'}
				shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- object characterized_object => {characterized_object.description = 'design layer'}
DESIGN_LAYER_TYPE_SPECIFIC_PADSTACK_DEFINITION	design_layer_type_specific_- padstack_definition	210		stratum_technology design_layer_type_specific_padstack_definition <= padstack_definition
ELECTRICAL_ISOLATION_REMOVAL_TEMPLATE	electrical_isolation_removal_- template_definition	210		electrical_isolation_removal_template_- definition <= part_template_definition <= shape_aspect (shape_aspect.description = 'material removal feature template')

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
electrical_isolation_removal_- template to layout_spacing_requirement_- occurrence (as electrical_isolation_spacing_- requirement)	PATH			<pre> electrical_isolation_removal_template_- definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition property_definition.relationship property_definition.relationship.related_- property_definition -> property_definition => requirements_property => grouped_requirements_property {grouped_requirements_property <= group group.name = 'layout spacing requirements property'} </pre>
FOOTPRINT_DEFINITION	footprint_definition	210		<pre> footprint_definition <= product_definition {[product_definition product_- definition.formation -> product_definition.formation product_definition.formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition product_definition.frame_of_reference -> </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'layout design usage'] }
footprint_definition to assembly_group_component (as reference_breakout)	PATH			(footprint_definition <= product_definition <- product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'reference_breakout' } product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => assembly_group_component_definition) (footprint_definition <= product_definition = characterized_product_ definition characterized_product_definition = characterized_definition characterized_definition <- property_definition.definition property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'reference_breakout' } property_definition_relationship property_definition_relationship.related_ property_definition property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> shape_aspect =>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				component_shape_aspect => assembly_group_component_shape_aspect)
LAYOUT_MACRO_DEFINITION_TERMINAL_TO_USAGE_TERMINAL_ASSIGNMENT	layout_macro_definition_terminal_to_usage_terminal_assignment	210		{[[layout_macro_definition_terminal_to_usage_terminal_assignment shape_aspect] [[layout_macro_definition_terminal_to_usage_terminal_assignment shape_aspect]relationship]] shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => printed_part_template_terminal
layout_macro_definition_terminal_to_usage_terminal_assignment to printed_part_template_terminal (as defined_template_feature)	PATH			
layout_macro_definition_terminal_to_usage_terminal_assignment to component_termination_passage_join_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_terminal_assignment shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> aspect -> {shape_aspect shape_aspect.description = 'component termination passage join terminal'} shape_aspect => component_terminal
layout_macro_definition_terminal_to_usage_terminal_assignment to conductive_interconnect_element_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_terminal_assignment shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> aspect -> {shape_aspect shape_aspect.description = 'conductive interconnect element terminal'} shape_aspect => component_terminal

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_macro_definition_- terminal_to_usage_terminal_- assignment to land_join_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'land join terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to non_functional_land_join_- terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'non functional land join terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to printed_component_join_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'printed component join terminal'} shape_aspect => component_terminal

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_macro_definition_- terminal_to_usage_terminal_- assignment to via_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'via terminal'} shape_aspect => component_terminal
layout_macro_definition_- terminal_to_usage_terminal_- assignment to embedded_component_terminal (as assigned_design_object)	PATH			layout_macro_definition_terminal_to_usage_- terminal_assignment <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect.description = 'embedded component terminal'} shape_aspect => component_terminal
LAYOUT_MACRO_FLOOR_- PLAN_TEMPLATE	layout_macro_floor_plan_- template	210		layout_macro_floor_plan_template <= part_template_definition <= {shape_aspect shape_aspect.description = 'printed part template'}
layout_macro_floor_plan_- template to layout_macro_definition_- terminal_to_usage_terminal_- assignment (as access_maps)	PATH			layout_macro_floor_plan_template <= part_template_definition <= shape_aspect <= shape_aspect_relationship.relatng_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'access maps'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => layout_macro_definition_terminal_to_usage_- terminal_assignment

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
MATERIAL_ADDITION_FEATURE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'material addition feature template'}
material_addition_feature_- template to stratum_technology (as of_stratum_technology)	PATH			part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'}
				shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- object characterized_object => stratum_technology
MATERIAL_REMOVAL_FEATURE_TEMPLATE	part_template_definition	210		part_template_definition <= {shape_aspect shape_aspect.description = 'material removal feature template'}
material_removal_feature_- template to stratum_technology (as of_stratum_technology)	PATH			part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'}
				shape_aspect_relationship.relatng_shape_- aspect ->

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ object characterized_object => stratum_technology
PADSTACK_DEFINITION	padstack_definition	210		padstack_definition <= product_definition {[product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <= product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout design usage']}]
padstack_definition to reference_packaged_part_ interconnect_implementation (as reference_implementation)	PATH			padstack_definition <= product_definition <= product_definition_relationship.related_ product_definition {product_definition_relationship product_definition_relationship.name = 'reference_implementation'} product_definition_relationship.related_ product_definition ->

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit => reference_packaged_part_interconnect_implementation
padstack_definition to length_data_element (as teardrop_taper_start_distance)	PATH			padstack_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'teardrop taper start distance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
padstack_definition to length_data_element (as teardrop_taper_end_distance)	PATH			padstack_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <=

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'teardrop taper end distance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_start_distance)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item {representation_item.name = 'snowball start distance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_end_distance)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'snowball end distance'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_initial_spacing)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'snowball initial spacing'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_end_spacing)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item {representation_item.name = 'snowball end spacing'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_initial_radius)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'snowball initial radius'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
padstack_definition to length_data_element (as snowball_end_radius)	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} representation.items[i] -> representation_item {representation_item.name = 'snowball end radius'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
snowball_quantity_per_trace	PATH			<pre> padstack_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation.name = 'padstack parametric data'} </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item {representation_item.name = 'snowball quantity per trace'} representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.unit_component -> unit unit = named_unit named_unit => context_dependent_unit} measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure </pre>
PADSTACK_LOCATION_IN_- FOOTPRINT_DEFINITION	assembly_component_usage	44		<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'padstack location in footprint definition'} </pre>
padstack_location_in_footprint_- definition to footprint_definition (as footprint)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relationship.relati- product_definition -> product_definition => footprint_definition </pre>
padstack_location_in_footprint_- definition to package_terminal (as reference_package_terminal)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_- definition_relationship characterized_definition = characterized_- product_definition characterized_definition <- </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'reference_package_terminal'} property_definition_relationship.relatiing_- property_definition -> property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => package_terminal </pre>
padstack_location_in_footprint_- definition to padstack_definition (as padstack)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => padstack_definition </pre>
PART_TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[product_definition product_definition.formatation -> product_definition.formatation product_definition.formatation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'template definition']]</pre>
part_template to analytical_model (as part_template_analytical_model)	PATH		333, 340	<pre>part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation => analytical_model</pre>
part_template to analytical_representation (as analytical_part_template_- representation)	PATH			<pre>part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition</pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {[representation => analytical_model] [representation representation.name = 'part template analytical_model']}] representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => analytical_representation
part_template to characteristic (as physical_characteristic)	PATH		333, 340, 109	part_template_definition <= {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.frame_of_reference -> product_definition_context <=

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'template definition' shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type} </pre>
part_template to coordinated_characteristic (as physical_characteristic)	PATH			<pre> part_template_definition <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> (characterized_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'template definition') shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => coordinated_representation_item part_template_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition
part_template to template_definition (as associated_definition)	PATH			part_template_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_product_definition characterized_product_definition = product_ definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'} </pre>
PART_TEMPLATE_LOCATION_IN_PADSTACK_DEFINITION	assembly_component_usage	44		<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'part template location in padstack definition'} </pre>
reference_designation	assembly_component_usage.reference_designator	44		
part_template_location_in_padstack_definition to template_definition (as template)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'} product_definition </pre>
part_template_location_in_padstack_definition to padstack_definition (as padstack)	PATH			<pre> assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition => padstack_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PART_TEXT_TEMPLATE	part_text_template_definition	210	333, 340	part_text_template_definition <= part_template_definition <= shape_aspect
part_text_template to length_data_element (as maximum_font_horizontal_extent)	PATH			part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum font horizontal extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
part_text_template to length_data_element (as maximum_font_vertical_extent)	PATH		333, 340	part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum font vertical extent'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
part_text_template to angle_data_element (as slant_angle)	PATH			part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'font slant angle'} representation_item => measure_representation_item <=

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit => plane_angle_measure_with_unit
part_text_template to length_data_element (as stroke_width)	PATH			part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'font stroke width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
part_text_template to registered_font (as font)	PATH		356	part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ - representation -> representation representation.items[i] -> {representation_item representation_item.name = 'registered font'} representation_item => geometric_representation_item => text_literal text_literal.font -> font_select font_select = externally_defined_text_font externally_defined_text_font </pre>
part_text_template to font_definition (as font)	PATH		333, 340, 356	<pre> part_text_template_definition <= part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ - representation -> representation representation.items[i] -> representation_item => geometric_representation_item => text_literal text_literal.font -> font_select font_select = externally_defined_text_font externally_defined_text_font </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
SPECIAL_SYMBOL_PART_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'special symbol part template'}
special_symbol_part_template to stratum_technology (as of_stratum_technology)	PATH			part_template_definition <= shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'}
				shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- object characterized_object => stratum_technology
STRATUM_FEATURE_TEMPLATE	part_template_definition	210		part_template_definition <= {shape_aspect (shape_aspect.description = 'stratum feature template') (shape_aspect.description = 'default attachment size based') (shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based')}

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'default unsupported passage based') (shape_aspect.description = 'default component termination passage based')}
stratum_feature_template to stratum_technology (as of_stratum_technology)	PATH			part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'of stratum technology'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- object characterized_object => stratum_technology
STRATUM_OCCURRENCE_- SPECIFIC_PADSTACK_DEFINI- TION	stratum_occurrence_specific_- padstack_definition	210		stratum_occurrence_specific_padstack_- definition <= padstack_definition
stratum_occurrence_specific_- padstack_definition to stratum (as reference_stratum)	PATH			stratum_occurrence_specific_padstack_- definition <= padstack_definition <= product_definition <- product_definition_relationship.related_- product_definition {product_definition_relationship product_definition_relationship.name = 'reference stratum'}

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.relati- product_definition -> product_definition => stratum
STRATUM_TYPE_INDEPENDENT_PADSTACK_DEFINITION	stratum_type_independent_ padstack_definition	210		stratum_type_independent_padstack_definition <= padstack_definition
TEMPLATE_DEFINITION	product_definition	41		product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'template definition'}
TEMPLATE_STRATUM_ASSIGNMENT	template_stratum_assignment	210		template_stratum_assignment <= shape_aspect_relationship
template_stratum_assignment to stratum (as bound_stratum)	PATH			template_stratum_assignment <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> product_definition => stratum
template_stratum_assignment to stratum_feature_template (as bound_template)	PATH			template_stratum_assignment <= shape_aspect_relationship shape_aspect_relationship.relati- shape_aspect -> {shape_aspect (shape_aspect.description = 'stratum feature template') (shape_aspect.description = 'default attachment size based')}

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'default attachment size and component termination passage based') (shape_aspect.description = 'default attachment size and via based') (shape_aspect.description = 'default via based') (shape_aspect.description = 'default unsupported passage based') (shape_aspect.description = 'default component termination passage based')}} shape_aspect => part_template_definition</pre>
template_stratum_assignment to material_addition_feature_- template (as bound_template)	PATH			<pre>template_stratum_assignment <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material addition shape'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect [{shape_aspect shape_aspect.description = 'material addition feature template'}]}</pre>
template_stratum_assignment to material_removal_feature_- template (as bound_template)	PATH			<pre>template_stratum_assignment <= shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material removal shape'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect</pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[[shape_aspect shape_aspect.description = 'material removal feature template']]
THERMAL_ISOLATION_RE- MOVAL_TEMPLATE	thermal_isolation_removal_ template_definition	210		thermal_isolation_removal_template_definition <= part_template_definition <= shape_aspect (shape_aspect.description = 'material removal feature template')
TRACE_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'trace template'}
trace_template to curve_style (as trace_style)	PATH		333, 340	part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'curve style parameters'}
USAGE_CONCEPT_USAGE_ RELATIONSHIP	mapped_item	43		{mapped_item {mapped_item <= representation_item representation_item.name = 'ucur'} mapped_item.mapping_source -> representation_map.mapping_origin -> representation_item =>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_concept_usage_relationship to axis_placement (as associated_usage_placement)				geometric_representation_item => placement}
usage_concept_usage_relationship to cartesian_transformation_ operator_2d (as associated_usage_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement
usage_concept_usage_relationship to component_termination_ passage_template_terminal (as associated_usage)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => cartesian_transformation_operator_2d
usage_concept_usage_relationship to component_termination_ passage_template_terminal (as associated_usage)	PATH			mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal')}}

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_concept_usage_relationship to shape_definition (as associated_shape_definition)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation
usage_concept_usage_relationship to land_template_terminal (as associated_usage)	PATH		333	mapped_item <= representation_item <= representation.items[i] {representation => shape_representation} representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')}} shape_aspect => land_template_terminal
usage_concept_usage_relationship to non_feature_shape_element (as associated_usage)	PATH			mapped_item <= representation_item <= representation.items[i] {representation => shape_representation} representation <=

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.product_definitional = .FALSE.} </pre>
usage_concept_usage_relationship to non_feature_shape_definition (as associating_non_feature_ - usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {{representation => shape_representation} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.product_definitional = .FALSE.}} </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_concept_usage_relationship to part_template_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
usage_concept_usage_relationship to part_template_non_planar_2d_ - shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'part template non planar 2d shape'] [representation <- </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation </pre>
usage_concept_usage_relationship to part_template_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
usage_concept_usage_relationship to physical_unit_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation } </pre>
usage_concept_usage_relationship to physical_unit_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape'] } </pre>

Table 78 – Mapping table for part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit}] representation => shape_representation </pre>
usage_concept_usage_relationship to printed_part_template_terminal (as associated_usage)	PATH		333	<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect </pre>

Table 78 – Mapping table for part_template UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')} shape_aspect => printed_part_template_terminal
usage_concept_usage_relationship to via_template_terminal (as associated_usage)	PATH			mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'via template terminal'}

Table 79 – Mapping table for part_template_shape UoF

Application element	AIM element	Source	Rules	Reference path
FOOTPRINT_DEFINITION_- SHAPE	shape_representation	41		<pre> {shape_representation <= [representation representation.name = 'planar projected shape'] [representation representation.description = 'footprint definition shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => footprint_definition]} </pre>
shape_environment	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item
shape_material_condition	descriptive_representation_item	45		shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item
footprint_definition_shape to footprint_definition (as shape_characterized_- footprint_definition)	PATH			shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => footprint_definition
PADSTACK_DEFINITION_- SHAPE	shape_representation	41		<pre> {shape_representation <= [representation representation.name = 'planar projected shape'] [representation representation.description = 'padstack definition shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => padstack_definition]} </pre>
shape_environment	descriptive_representation_item	45		shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item </pre>
padstack_definition_shape to padstack_definition (as shape_characterized_- padstack_definition)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition = characterized_ - product_definition characterized_product_definition = product_ - definition product_definition => padstack_definition </pre>
PADSTACK_LOCATION_IN_- FOOTPRINT_DEFINITION_- TRANSFORM	mapped_item	43		<pre> {mapped_item <= representation_item representation_item.name = 'padstack location in footprint definition transform'} mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => axis2_placement_2d]} representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape'] [representation representation.items[i] -> representation_item => geometric_representation_item => axis2_placement_2d] [representation <- property_definition_representation.used_ - representation property_definition_representation </pre>
padstack_location_in_footprint_- definition_transform to padstack_definition_shape (as padstack_shape)	PATH			

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => padstack_definition}} representation => shape_representation mapped_item mapped_item.mapping_target -> axis2_placement_2d </pre>
padstack_location_in_footprint_- definition_transform to axis_placement_2d (as transform)	PATH			
padstack_location_in_footprint_- definition_transform to padstack_location_in_footprint_- definition (as reference_location)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition_relationship {product_definition_relationship 'padstack_location_in_footprint_definition'} product_definition_relationship => product_definition_usage => assembly_component_usage </pre>
padstack_location_in_footprint_- definition_transform to footprint_definition_shape (as footprint_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition => footprint_definition[] representation => shape_representation </pre>
PART_TEMPLATE_3D_KEEP- OUT_SHAPE	shape_representation	41		<pre> {shape_representation <= [representation representation.name = '3d bound volume shape'] representation representation.description = 'keepout shape'] representation representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_ - space_dimension = 3] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[]} </pre>
constrained_design_object_cate- gory	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_ - representation property_definition_representation </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => {characterized_object (characterized_object.description = 'assembly module assembly component category') (characterized_object.description = 'component feature category') (characterized_object.description = 'assembly ee material category') (characterized_object.description = 'interconnect ee material category') (characterized_object.description = 'interconnect module assembly component category') (characterized_object.description = 'via category') (characterized_object.description = 'inter stratum feature category') (characterized_object.description = 'cutout category') (characterized_object.description = 'fill area category') (characterized_object.description = 'laminare component category') (characterized_object.description = 'stratum feature category')}} keepout_design_object_category shape_representation <= representation <- </pre>
shape_environment	descriptive_representation_item	45		

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')}} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')}} representation_item => descriptive_representation_item </pre>
part_template_3d_keepout_shape to part_template (as shape_characterized_part_- template)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition
part_template_3d_keepout_shape to cartesian_point (as centroid_location)	PATH			shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => geometric_representation_item => point => cartesian_point
part_template_3d_keepout_shape to ee_requirement_occurrence (as application_technology_- constraint)	PATH			shape_representation <= representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition <= property_definition_relationship.related_ property_definition {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property shape_representation <= representation <= representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation
part_template_3d_keepout_shape to external_definition (as shape_purpose)	PATH			

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition</pre>
physical_unit_3d_keepout_shape to predefined_3d_purpose (as shape_purpose)	PATH			<pre>shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined keepout shape purpose'} representation.items[i] -> representation_item {(representation_item.name = 'thermal') (representation_item.name = 'generic clearance') (representation_item.name = 'shock') (representation_item.name = 'vibration') (representation_item.name = 'electromagnetic compatibility')} representation_item => descriptive_representation_item</pre>
PART_TEMPLATE_3D_SHAPE	shape_representation	41	111, 343, 340, 333	<pre>{shape_representation <= [representation representation.name = '3d bound volume shape'] [representation representation.description != 'keepout shape'] [representation representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 3]</pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]]</pre>
shape_environment	descriptive_representation_item	45	109	<pre>shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item</pre>
shape_material_condition	descriptive_representation_item	45		<pre>shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation</pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')}} representative_item => descriptive_representation_item </pre>
part_template_3d_shape to part_template (as shape_characterized_part_- template)	PATH		340, 333	<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition </pre>
PART_TEMPLATE_LOCATION_IN_PADSTACK_DEFINITION_TRANSFORM	mapped_item	43		<pre> {mapped_item <= representation_item representation_item.name = 'part template location in padstack definition transform'} </pre>
part_template_location_in_- padstack_definition_transform to part_template_planar_shape (as template_shape)	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_representation_item => axis2_placement_2d[] representation_map.mapped_representation -> {[representation representation.name = 'planar projected shape']} [representation representation.items[i] -> representation_item => geometric_representation_item => axis2_placement_2d] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition characterized_product_definition = product_ - definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'}}}] </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation => shape_representation
part_template_location_in_- padstack_definition_transform to axis_placement_2d (as transform)	PATH			mapped_item mapped_item.mapping_target -> axis2_placement_2d
part_template_location_in_- padstack_definition_transform to part_template_location_in_- padstack_definition (as reference_location)	PATH			mapped_item <= representation_item <= representation.items[i] representation <= property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship {product_definition_relationship product_definition_relationship.name = 'part template_location_in_padstack_definition'} product_definition_relationship => product_definition_usage => assembly_component_usage
part_template_location_in_- padstack_definition_transform to padstack_definition_shape (as padstack_shape)	PATH			mapped_item <= representation_item <= representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <= property_definition_representation.used_- representation property_definition_representation

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => product_definition[] representation => shape_representation </pre>
PART_TEMPLATE_NON_PLA- NAR_2D_SHAPE	manifold_surface_shape_- representation	509	290, 234	<pre> {manifold_surface_shape_representation <= shape_representation <= representation representation.name = 'part template non planar 2d shape'} </pre>
shape_environment	descriptive_representation_item	45		<pre> manifold_surface_shape_representation <= shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shape_material_condition	descriptive_representation_item	45		<pre> manifold_surface_shape_representation <= shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} descriptive_representation_item </pre>
part_template_non_planar_2d_- shape to axis_placement_3d (as transformation)	PATH			<pre> manifold_surface_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>
part_template_non_planar_2d_- shape to part_template (as shape_characterized_part_- template)	PATH			<pre> manifold_surface_shape_representation <= shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition
PART_TEMPLATE_OR_PHYSICAL_UNIT_2D_SHAPE_REPRESENTATION_RELATIONSHIP	shape_representation_relationship	41		{shape_representation_relationship <= representation_relationship [representation_relationship representation_relationship.name = 'part template or physical unit planar shape representation relationship']} shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape'] [representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition ->
part_template_or_physical_unit_2d_shape_representation - relationship to part_template_planar_shape (as principal_part_shape)	PATH			

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
part_template_or_physical_unit_- 2d_shape_representation_- relationship to physical_unit_planar_shape (as principal_part_shape)	PATH			<pre> shape_representation relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition definition product_definition => physical_unit[] representation => shape_representation </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_template_or_physical_unit_- 2d_shape_representation_- relationship to part_template_planar_shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
part_template_or_physical_unit_- 2d_shape_representation_- relationship to physical_unit_planar_shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => physical_unit[] representation => shape_representation
PART_TEMPLATE_OR_PHYSICAL_UNIT_3D_SHAPE_REPRESENTATION_RELATIONSHIP	shape_representation_relationship	41		{shape_representation_relationship <= representation_relationship [representation_relationship representation_relationship.name = 'part template or physical unit 3d shape representation relationship']} shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition ->
part_template_or_physical_unit_3d_shape_representation - relationship to part_template_3d_shape (as principal_part_shape)	PATH			

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_non_planar_2d_- shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
part_template_or_physical_unit_ 3d_shape_representation_ relationship to physical_unit_3d_shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_3d_shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_non_planar_2d_- shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_- representation property_definition_representation.definition -> property_definition </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
part_template_or_physical_unit_ 3d_shape_representation_ relationship to physical_unit_3d_shape (as related_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PART_TEMPLATE_PLANAR_- KEEPOUT_SHAPE	shape_representation	41		<pre> {shape_representation <= [representation representation.name = 'planar projected shape'] [representation representation.description = 'keepout shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]]} </pre>
constrained_design_object_category	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - object characterized_object => {characterized_object (characterized_object.description = 'assembly module assembly component category')} </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (characterized_object.description = 'component feature category') (characterized_object.description = 'assembly ee material category') (characterized_object.description = 'interconnect ee material category') (characterized_object.description = 'interconnect module assembly component category') (characterized_object.description = 'via category') (characterized_object.description = 'inter stratum feature category') (characterized_object.description = 'cutout category') (characterized_object.description = 'fill area category') (characterized_object.description = 'laminated component category') (characterized_object.description = 'stratum feature category') keepout_design_object_category </pre>
shape_environment	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')}} representation_item => descriptive_representation_item </pre>
part_template_planar_keepout_- shape to part_template (as shape_characterized_part_- template)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition </pre>
part_template_planar_keepout_- shape to cartesian_point (as centroid_location)	PATH			<pre> shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_representation_item => point => cartesian_point </pre>
part_template_planar_keepout_shape to ee_requirement_occurrence (as application_technology_constraint)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.related_ property_definition -> property_definition => requirements_property </pre>
part_template_keepout_shape to external_definition (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition </pre>
physical_unit_3d_keepout_shape to predefined_3d_purpose (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined keepout shape purpose'} </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> representation_item {(representation_item.name = 'thermal') (representation_item.name = 'generic clearance') (representation_item.name = 'shock') (representation_item.name = 'vibration') (representation_item.name = 'electromagnetic compatibility')} representation_item => descriptive_representation_item </pre>
PART_TEMPLATE_PLANAR_SHAPE	shape_representation	41	111, 235, 340, 333	<pre> {shape_representation <= [representation representation.name = 'planar projected shape']} [representation representation.description != 'keepout shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} </pre>
shape_environment	descriptive_representation_item	45	109	<pre> shape_representation <- representation <- </pre>

Table 79 – Mapping table for part_template_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')}} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')}} representation_item => descriptive_representation_item </pre>
part_template_planar_shape to part_template (as shape_characterized_part_- template)	PATH		340, 333	<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 79 – Mapping table for part_template_shape UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition

Table 80 – Mapping table for physical_network UoF

Application element	AIM element	Source	Rules	Reference path
PARTIALLY_ROUTED_PHYSICAL_NETWORK	physical_network	210		physical_network <= shape_aspect {shape_aspect.description = 'partially routed physical_network'}
partially_routed_physical_network to unrouted_join_relationship (as unrouted_connectivity_ requirement_element)	PATH			physical_network <= shape_aspect <= shape_aspect_relationship.relatiing_shape_ aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'unrouted connectivity requirement'}
PHYSICAL_COMPONENT_ TERMINAL	(component_terminal) (component_interface_terminal)	210 210		shape_aspect => {shape_aspect shape_aspect.name = 'unrouted join'} join_shape_aspect (component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
PHYSICAL_NETWORK	physical_network	210		physical_network <= shape_aspect
name	shape_aspect.name	41		physical_network <= shape_aspect shape_aspect.name

Table 80 – Mapping table for physical_network UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_network to interconnect_module (as current_associated_definition)	PATH			<pre> physical_network <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) physical_network <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'prior associated_definition'} shape_aspect_relationship.related_shape_- aspect -> </pre>
physical_network to interconnect_module (as prior_associated_definition)	PATH			

Table 80 – Mapping table for physical_network UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) physical_network <= shape_aspect <- shape_aspect_relationship.relatng_shape_ aspect shape_aspect_relationship {shape_aspect_relationship
physical_network to join_relationship (as interconnect_module - network_topology_element)	PATH			

Table 80 – Mapping table for physical_network UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'network topology' shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.name = 'constrained intra layer join') (shape_aspect.name = 'inter stratum join') (shape_aspect.name = 'intra stratum join')} shape_aspect => join_shape_aspect </pre>
physical_network to laminate - component_join_terminal (as connectivity_requirement - element)	PATH			<pre> physical_network <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'connectivity requirement'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'component termination passage join terminal') (shape_aspect.description = 'land join terminal') (shape_aspect.description = 'non functional land join terminal') (shape_aspect.description = 'printed component join terminal')} shape_aspect => component_terminal </pre>
physical_network to part_- connected_terminals_definition (as reference_connected_terminals)	PATH			<pre> physical_network <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship </pre>

Table 80 – Mapping table for physical_network UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.name = 'reference connected_terminals' shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_connected_terminals_definition

Table 81 – Mapping table for physical_unit UoF

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT_CATEGORY	characterized_product_category	210		characterized_product_category <= characterized_product_category <= [product_category] [characterized_object]
description	product_category.description	41		characterized_product_category <= product_category product_category.description
name	product_category.name	41		characterized_product_category <= product_category product_category.name
ee_product_category to model_parameter (as category_parameter)	PATH		340, 109	characterized_product_category <= characterized_object characterized_definition = characterized_- object characterized_definition characterized_definition <- property_definition.definition property_definition {property_definition.description = 'product category_parameter'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> representation_item representation_item => model_parameter

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product_category to parameter_assignment (as category_specific_values)	PATH			<pre> characterized_product_category <= characterized_object characterized_definition = characterized_ object characterized_definition characterized_definition <- property_definition.definition property_definition {property_definition.description = 'product category values'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation => parameter_assignment_representation </pre>
ee_product_category to ee_document (as reference_document)	PATH			<pre> characterized_product_category <= characterized_object document_assigned_item = characterized_object document_assigned_item document_assigned_item <- applied_document_reference.items[i] applied_document_reference applied_document_reference <= document_reference {document_reference.role -> object_role object_role.name = 'product category reference document'} document_reference.assigned_document -> (document) (document => ee_specification) </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT_CATEGORY_- RELATIONSHIP	product_category_relationship	41		product_category_relationship
ee_product_category_relationship to ee_product_category (as category)	PATH			product_category_relationship product_category_relationship.category -> product_category => characterized_product_category
ee_product_category_relationship to ee_product_category (as sub_category)	PATH			product_category_relationship product_category_relationship.sub_category -> product_category => characterized_product_category
EE_PRODUCT_RELATED_- PRODUCT_CATEGORY	product_related_characterized_- product_category	210		product_related_characterized_product_category <= characterized_product_category <= product_category {product_category => product_related_product_category}
ee_product_related_product_- category to ee_product (as of_product)	PATH			product_related_characterized_product_category <= product_related_product_category product_related_product_category.products[i] -> product
EE_PRODUCT_SPECIFIC_PA- RAMETER_VALUE_ASSIGN- MENT	product_specific_parameter_- value_assignment	210		product_specific_parameter_value_assignment <= product_related_characterized_product_category <= characterized_product_category <= product_category {product_category => product_related_product_category}
ee_product_specific_parameter_- value_assignment to parameter_assignment (as assigned_parameter)	PATH		340, 109	product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category <= product_category => characterized_product_category <= characterized_object

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ object characterized_definition characterized_definition <- property_definition.definition property_definition {property_definition.description = 'product category parameter'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => (model_parameter) (measure_representation_item) (descriptive_representation_item)
ee_product_specific_parameter_ value_assignment to ee_product_related_product_ category (as associated_product_category)	PATH			product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category <= product_category <- product_category_relationship.sub_category product_category_relationship {product_category_relationship.description = 'product value assignment'} product_category_relationship.category ->

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_category => product_related_product_category => product_related_characterized_product_category <= product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category product_related_product_category.products[i] -> product
ee_product_specific_parameter_- value_assignment to ee_product (as of_product)	PATH			
PHYSICAL_UNIT	(physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)	210 210 210	338	(physical_unit <= product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')}
physical_unit to analytical_representation (as analytical_physical_unit_- representation)	PATH		338	(physical_unit <= (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <=)

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation => analytical_model} representation <= representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => analytical_representation
physical_unit to characteristic (as physical_device_characteristic)	PATH			(physical_unit <= externally_defined_physical_unit <= physical_unit <= (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition {property_definition property_definition.description = 'physical device characteristic'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type}
physical_unit to coordinated_characteristic (as physical_device_coordinated_ characteristic)	PATH		338, 342	(physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition {property_definition property_definition.description = 'physical device coordinated characteristic'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => coordinated_representation_item
physical_unit to ee_requirement_occurrence (as design_requirement)	PATH		338	(physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship product_definition_relationship.related_ product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'}

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition => requirements_property (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <= property_definition.definition property_definition <= property_definition.relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'physical unit technology'} property_definition_relationship.related_ property_definition -> property_definition {property_definition
physical_unit to technology (as unit_technology)	PATH		338	

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.name = 'unit technology'
PHYSICAL_UNIT_DESIGN_VIEW	physical_unit	210	338	<pre> physical_unit <= product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} </pre>
physical_unit_design_view to physical_unit_usage_view (as usage_view)	PATH		338	<pre> physical_unit <= product_definition <- product_definition_relationship.related_ - product_definition {product_definition_relationship product_definition_relationship.name = 'design usage'} product_definition_relationship product_definition_relationship.related_ - product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design usage'} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
PHYSICAL_UNIT_RELATIONSHIP	product_definition_relationship	41	244	<pre> product_definition_relationship product_definition_relationship.name = 'physical unit relationship' </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_unit_relationship to physical_unit (as related_unit)	PATH		338	<pre> product_definition_relationship product_definition_relationship.related_ - product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
physical_unit_relationship to physical_unit (as relating_unit)	PATH			<pre> product_definition_relationship product_definition_relationship.relatin_ - product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_UNIT_USAGE_- VIEW	(physical_unit) (externally_defined_physical_unit) (library_defined_physical_unit)	210 210 210	338	(physical_unit <= product_definition) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage'}
TECHNOLOGY	material_property	45	358	material_property material_property <= property_definition {property_definition.name = 'unit technology'}
technology to external_definition (as classification)	PATH			material_property material_property <= property_definition <- property_definition.relationship.related_ property_definition property_definition.relationship {property_definition.relationship property_definition.relationship.name = 'classification'} property_definition.relationship.relati- ng_ property_definition -> property_definition property_definition.definition -> characterized_definition

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ - object characterized_object characterized_object => external_definition
technology to ee_material (as material_product_system)	PATH			material_property <= property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition = product_ - definition product_definition {product_definition product_definition.description = 'material product system'} characterized_product_definition = product_ - definition characterized_definition = characterized_ - product_definition characterized_definition <- material_designation.definitions
technology to process_specification (as process_technology)	PATH			material_property material_property <= property_definition property_definition<- property_definition.representation.definition property_definition.representation property_definition.representation.used_ - representation-> representation document_assigned_item = representation document_assigned_item document_assigned_item<- applied_document_reference.items[i] applied_document_reference

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> applied_document_reference<= document_reference document_reference.assigned_document-> document {document.kind -> document_type document_type.product_data_type = 'process specification'} document => ee_specification </pre>
USAGE_CONCEPT_USAGE_- RELATIONSHIP	mapped_item	43		<pre> {mapped_item {mapped_item <= representation_item representation_item.name = 'ucur'} mapped_item.mapping_source -> representation_map representation_map.mapping_origin -> representation_item => geometric_representation_item => placement} mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => cartesian_transformation_operator_2d mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation </pre>
usage_concept_usage_relationship to axis_placement (as associated_usage_placement)	PATH			
usage_concept_usage_relationship to cartesian_transformation_- operator_2d (as associated_usage_placement)	PATH			
usage_concept_usage_relationship to shape_definition (as associated_shape_definition)	PATH			

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
usage_concept_usage_relationship to non_feature_shape_element (as associated_usage)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.} </pre>
usage_concept_usage_relationship to non_feature_shape_definition (as associating_non_feature_ - usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {{representation => shape_representation} representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}} </pre>
usage_concept_usage_relationship to part_feature (as associated_usage)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface')} </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
usage_concept_usage_relationship to part_template_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
usage_concept_usage_relationship to part_template_non_planar_2d_ - shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'part template non planar 2d shape'] [representation <- </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation </pre>
usage_concept_usage_relationship to part_template_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition </pre>

Table 81 – Mapping table for physical_unit UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
usage_concept_usage_relationship to physical_unit_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit]} representation => shape_representation </pre>
usage_concept_usage_relationship to physical_unit_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape'] </pre>

Table 81 – Mapping table for physical_unit UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				[representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit}} representation => shape_representation

Table 82 – Mapping table for physical_unit_analytical_representation UoF

Application element	AIM element	Source	Rules	Reference path
ANALYTICAL_REPRESENTATION	analytical_representation	210		analytical_representation <= representation
analytical_representation_name	representation.name	43		analytical_representation <= representation representation.name
analytical_representation to analytical_model (as model_representation)	PATH			analytical_representation <= representation <- representation_relationship representation_relationship representation_relationship.rep_1 -> representation => analytical_model
analytical_representation to parameter_assignment (as model_parameter_assignment)	PATH			analytical_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => parameter_assignment_representation
CONNECTION_ZONE_-MODEL_PORT_ASSIGNMENT	property_definition_representation	41		property_definition_representation
connection_zone_model_port_-assignment to connection_zone (as source)	PATH			property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition

Table 82 – Mapping table for physical_unit_analytical_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.description = 'connection zone'}
connection_zone_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation property_definition_representation.used_ representation -> representation => analytical_model_port
PART_FEATURE_MODEL_- PORT_ASSIGNMENT	property_definition_representation	41		{property_definition_representation property_definition_representation.name = 'part feature model port assignment'}
part_feature_model_port_- assignment to analytical_model_port (as target)	PATH		342	property_definition_representation {property_definition_representation.name = 'part feature model port assignment'} property_definition_representation.used_ representation -> representation => analytical_model_port
part_feature_model_port_- assignment to part_feature (as source)	PATH			property_definition_representation {property_definition_representation.name = 'part feature model port assignment'} property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {[shape_aspect => composite_shape_aspect] [shape_aspect]

Table 82 – Mapping table for physical_unit_analytical_representation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.description = 'part group feature'] (shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => physical_unit_datum_feature => primary_orientation_feature) (shape_aspect => physical_unit_datum_feature => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect =>

Table 82 – Mapping table for physical_unit_analytical_representation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)}

Table 83 – Mapping table for physical_unit_shape UoF

Application element	AIM element	Source	Rules	Reference path
PART_TEMPLATE_OR_PHYSICAL_UNIT_2D_SHAPE_REPRESENTATION_RELATIONSHIP	shape_representation_relationship	41		{shape_representation_relationship <= representation_relationship [representation_relationship representation_relationship.name = 'part template or physical unit planar shape representation relationship']}
part_template_or_physical_unit_2d_shape_representation_relationship to part_template_planar_shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation
part_template_or_physical_unit_2d_shape_representation_relationship to physical_unit_planar_shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit[] representation => shape_representation </pre>
<pre> part_template_or_physical_unit_- 2d_shape_representation_- relationship to part_template_planar_shape (as related_part_shape) </pre>	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
<pre> part_template_or_physical_unit_- 2d_shape_representation_- relationship to physical_unit_planar_shape (as related_part_shape) </pre>	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition definition product_definition => physical_unit[] representation => shape_representation
PART_TEMPLATE_OR_PHYSICAL_UNIT_3D_SHAPE_REPRESENTATION_RELATIONSHIP	shape_representation_relationship	41		{shape_representation_relationship <= representation_relationship [representation_relationship representation_relationship.name = 'part template or physical unit 3d shape representation relationship'] }
part_template_or_physical_unit_3d_shape_representation_relationship to part_template_3d_shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition ->

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_non_planar_2d_- shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
part_template_or_physical_unit_ 3d_shape_representation_ relationship to physical_unit_3d_shape (as principal_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_3d_shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
part_template_or_physical_unit_- 3d_shape_representation_- relationship to part_template_non_planar_2d_- shape (as related_part_shape)	PATH			<pre> shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = 'part template non planar 2d shape'] [representation <- property_definition_representation.used_- representation property_definition_representation.definition -> property_definition </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation
part_template_or_physical_unit_ 3d_shape_representation_ relationship to physical_unit_3d_shape (as related_part_shape)	PATH			shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_UNIT_3D_KEEP- OUT_SHAPE	shape_representation	41		<pre> shape_representation <= representation {[representation.name = '3d bound volume shape'] [(representation.description = 'keepout shape') (representation.description = 'package keepout shape')] [representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 3] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => physical_unit]]} </pre>
constrained_design_object_cate- gory	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition characterized_definition -> characterized_definition characterized_definition = characterized_ object characterized_object => {characterized_object (characterized_object.description = 'assembly module assembly component category') (characterized_object.description = 'component feature category') (characterized_object.description = 'assembly ee material category') (characterized_object.description = 'interconnect ee material category') (characterized_object.description = 'interconnect module assembly component category') (characterized_object.description = 'via category') (characterized_object.description = 'inter stratum feature category') (characterized_object.description = 'cutout category') (characterized_object.description = 'fill area category') (characterized_object.description = 'laminated component category') (characterized_object.description = 'stratum feature category')}} keepout_design_object_category </pre>
shape_environment	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item </pre>
physical_unit_3d_keepout_shape to external_definition (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition </pre>
physical_unit_3d_keepout_shape to predefined_3d_purpose (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined keepout shape purpose'} representation.items[i] -> representation_item {(representation_item.name = 'thermal') (representation_item.name = 'generic clearance') (representation_item.name = 'shock') (representation_item.name = 'vibration') (representation_item.name = 'electromagnetic compatibility')} representation_item => descriptive_representation_item </pre>
physical_unit_3d_keepout_shape to ee_requirement_occurrence (as application_technology_- constraint)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition <- property_definition_relationship.related_- property_definition </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.relatng_- property_definition -> property_definition => requirements_property </pre>
<pre> physical_unit_3d_keepout_shape to physical_unit (as shape_characterized_- physical_unit) </pre>	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_unit_3d_keepout_shape to cartesian_point (as centroid_location)	PATH			<pre> shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => geometric_representation_item => point => cartesian_point </pre>
PHYSICAL_UNIT_3D_SHAPE	shape_representation	41	111, 242, 343, 338	<pre> shape_representation <= representation {[representation.name = '3d bound volume shape']} [representation.description != 'keepout shape'] [representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 3] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition => physical_unit]} </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shape_environment	descriptive_representation_item	45	109	<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item </pre>
physical_unit_3d_shape to external_definition (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation {representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition </pre>
physical_unit_3d_shape to predefined_3d_purpose (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined shape purpose'} representation.items[i] -> representation_item {(representation_item.name = 'analysis input') (representation_item.name = 'analysis output') (representation_item.name = 'shock analysis input') (representation_item.name = 'shock analysis output') (representation_item.name = 'design') (representation_item.name = 'vibration analysis input') (representation_item.name = 'vibration analysis output') (representation_item.name = 'electromagnetic compatibility analysis input') (representation_item.name = 'electromagnetic compatibility analysis output') (representation_item.name = 'thermal analysis input') </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(representation_item.name = 'thermal analysis output')} representation_item => descriptive_representation_item
physical_unit_3d_shape to ee_requirement_occurrence (as application_technology_- constraint)	PATH			shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property
physical_unit_3d_shape to physical_unit (as shape_characterized_- physical_unit)	PATH		338	shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition {product_definition

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage') product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
physical_unit_3d_shape to cartesian_point (as centroid_location)	PATH			<pre> shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => geometric_representation_item => point => cartesian_point </pre>
PHYSICAL_UNIT_PLANAR_- KEEPOUT_SHAPE	shape_representation	41		<pre> shape_representation <= representation {[(representation.description = 'keepout shape') (representation.description = 'package keepout shape')]} [representation <- </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[]
constrained_design_object_category	PATH			shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => {characterized_object (characterized_object.description = 'assembly module assembly component category') (characterized_object.description = 'component feature category') (characterized_object.description = 'assembly ee material category')}

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (characterized_object.description = 'interconnect ee material category') (characterized_object.description = 'interconnect module assembly component category') (characterized_object.description = 'via category') (characterized_object.description = 'inter stratum feature category') (characterized_object.description = 'cutout category') (characterized_object.description = 'fill area category') (characterized_object.description = 'laminated component category') (characterized_object.description = 'stratum feature category') keepout_design_object_category </pre>
shape_environment	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item </pre>
shape_location_with_respect_to_- seating_plane	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_relationship.rep_2 -> representation {representation.name = 'shape location'} representation.items[i] -> representation_item {(representation_item.name = 'above')} (representation_item.name = 'congruent') (representation_item.name = 'below')} representation_item => descriptive_representation_item </pre>
shape_material_condition	descriptive_representation_item	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition')} (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item </pre>
physical_unit_planar_keepout_- shape to external_definition (as shape_purpose)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition </pre>
physical_unit_planar_keepout_- shape to predefined_planar_purpose (as shape_purpose)	representation_item.name	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined keepout shape purpose'} representation.items[i] -> representation_item {(representation_item.name = 'thermal') (representation_item.name = 'generic clearance') (representation_item.name = 'shock') (representation_item.name = 'vibration') (representation_item.name = 'electromagnetic compatibility')} </pre>
physical_unit_planar_keepout_- shape to ee_requirement_occurrence (as application_technology_- constraint)	PATH			<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition <- property_definition_relationship.related_- property_definition </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.relatng_- property_definition -> property_definition => requirements_property </pre>
<pre> physical_unit_planar_keepout_- shape to physical_unit (as shape_characterized_- physical_unit) </pre>	PATH		<pre> shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>	

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_unit_planar_keepout_- shape to cartesian_point (as centroid_location)	PATH			<pre> shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => geometric_representation_item => point => cartesian_point </pre>
physical_unit_planar_keepout_- shape to length_data_element (as shape_distance_from_- seating_plane)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape location'} representation.items[i] -> representation_item {representation_item.name = 'shape distance from seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
PHYSICAL_UNIT_PLANAR_- SHAPE	shape_representation	41	111, 243, 338	<pre> shape_representation <= representation {[representation.name = 'planar projected shape']} [representation.description != 'keepout shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[]
shape_environment	descriptive_representation_item	45	109	shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape environment'} representation.items[i] -> representation_item {(representation_item.name = 'manufacturing') (representation_item.name = 'end user application')} representation_item => descriptive_representation_item
shape_location_with_respect_to_ seating_plane	descriptive_representation_item	45		shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape location'} representation.items[i] -> representation_item {(representation_item.name = 'above')}

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(representation_item.name = 'congruent') (representation_item.name = 'below') representation_item => descriptive_representation_item</pre>
shape_material_condition	descriptive_representation_item	45		<pre>shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape material condition'} representation.items[i] -> representation_item {(representation_item.name = 'maximum material condition') (representation_item.name = 'minimum material condition') (representation_item.name = 'nominal material condition')} representation_item => descriptive_representation_item</pre>
physical_unit_planar_shape to external_definition (as shape_purpose)	PATH			<pre>shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'externally defined shape purpose'} representation.items[i] -> representation_item => externally_defined_representation_item => external_definition</pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
physical_unit_planar_shape to length_data_element (as shape_distance_from_- seating_plane)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'shape location'} representation.items[i] -> representation_item {representation_item.name = 'shape distance from seating plane'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
physical_unit_planar_shape to predefined_planar_purpose (as shape_purpose)	representation_item.name	45		<pre> shape_representation <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation {representation.name = 'predefined shape purpose'} representation.items[i] -> representation_item {(representation_item.name = 'analysis input') (representation_item.name = 'analysis output') (representation_item.name = 'assembly symbol') (representation_item.name = 'shock analysis input') (representation_item.name = 'shock analysis output') (representation_item.name = 'design') (representation_item.name = 'design profile') (representation_item.name = 'vibration analysis input')} </pre>

Table 83 – Mapping table for physical_unit_shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(representation_item.name = 'vibration analysis output') (representation_item.name = 'electromagnetic compatibility analysis input') (representation_item.name = 'electromagnetic compatibility analysis output') (representation_item.name = 'thermal analysis input') (representation_item.name = 'thermal analysis output') (representation_item.name = 'physical extent')}</pre>
physical_unit_planar_shape to ee_requirement_occurrence (as application_technology_- constraint)	PATH			<pre>shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship.name = 'application technology constraint'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property</pre>
physical_unit_planar_shape to physical_unit (as shape_characterized_- physical_unit)	PATH		338	<pre>shape_representation <= representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition</pre>

Table 83 – Mapping table for physical_unit_shape UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) shape_representation shape_representation <= representation representation.items[i] -> representation_item representation_item => geometric_representation_item => point => cartesian_point
physical_unit_planar_shape to cartesian_point (as centroid_location)	PATH			

Table 84 – Mapping table for planned_characteristic UoF

Application element	AIM element	Source	Rules	Reference path
EVALUATED_CHARACTERIS- TIC	evaluated_characteristic	210	166	evaluated_characteristic <= [representation_relationship] [representation]
evaluation_result	representation_relationship.name	43		evaluated_characteristic <= representation_relationship representation_relationship.name {(representation_relationship.name = 'pass') (representation_relationship.name = 'fail') (representation_relationship.name = 'undefined')}}}
product_data_status	representation_- relationship.description	43		evaluated_characteristic <= representation_relationship representation_relationship.description = {(representation_relationship.description = 'valid') (representation_relationship.description = 'invalid') (representation_relationship.description = 'not evaluable') (representation_relationship.description = 'to be evaluated')}}}
evaluated_characteristic to planned_characteristic (as planned_product_data)	PATH			evaluated_characteristic <= representation_relationship {representation_relationship representation_relationship.name = 'planned product data'} representation_relationship.rep_1 -> representation {representation.name = 'planned characteristic'}}

Table 84 – Mapping table for planned_characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
evaluated_characteristic to coordinated_characteristic (as product_characterization_data)	PATH			evaluated_characteristic <= representation_relationship {representation_relationship {representation_relationship.name = 'product_characterization_data'}} representation_relationship.rep_2 -> representation representation.items[i] -> representation_item => coordinated_representation_item
EVALUATION_EE_PRODUCT_DEFINITION	evaluation_product_definition	210		evaluation_product_definition <= product_definition
evaluation_ee_product_definition to ee_product_definition (as evaluated_product_definition)	PATH			evaluation_product_definition <= product_definition <- product_definition.relationship.related_product_definition {product_definition_relationship.name = 'evaluated_product_definition'} product_definition.relationship.related_product_definition -> (product_definition => product_definition_with_associated_documents) (product_definition) (product_definition => evaluation_product_definition) (product_definition => physical_unit) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition => analytical_model_definition) (product_definition => footprint_definition)

Table 84 – Mapping table for planned_characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => stratum) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit) evaluation_product_definition <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation => evaluated_characteristic</pre>
evaluation_ee_product_definition to evaluated_characteristic (as evaluation_data)	PATH			
PLANNED_CHARACTERISTIC	representation	43		<pre>representation {representation.name = 'planned characteristic'}</pre>

Table 84 – Mapping table for planned_characteristic UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
planned_characteristic to coordinated_characteristic (as planned_coordinated_-characteristic)	PATH			representation.items[i] -> representation_item => coordinated_representation_item
planned_characteristic to requirement_allocation (as reference_requirement)	PATH			<pre> representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> (property_definition property_definition.definition -> product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'requirement_allocation'}) (property_definition property_definition.definition -> shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'requirement_allocation'}) (represented_definition = property_ - definition_relationship property_definition_relationship => {property_definition_relationship => requirement_allocation_group}) </pre>
RULE_BASED_EVALUATED_-CHARACTERISTIC	evaluated_characteristic	210		<pre> {[evaluated_characteristic <= representation_relationship] [representation_relationship representation_relationship.name = 'rule based evaluated_characteristic']} </pre>

Table 84 – Mapping table for planned_characteristic UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
rule_based_evaluated_- characteristic to ee_rule_version (as evaluating_rule)	PATH			[evaluated_characteristic <= representation]} evaluated_characteristic <= representation <- representation_relationship.rep_2 {representation_relationship representation_relationship.name = 'evaluating rule'} representation_relationship.rep_1 -> representation <- property_definition_representation.used_ representation property_definition_representation.definition -> represented_definition = property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition {product_definition => rule_definition} product_definition.formation -> product_definition_formation {product_definition_formation.description = 'rule version'}

Table 85 – Mapping table for planned_effectivity UoF

Application element	AIM element	Source	Rules	Reference path
PLANNED_DATE_EFFECTIVITY	dated_effectivity	41		
planned_date_effectivity to date (as end_date)	PATH			dated_effectivity.effectivity_end_date -> date_time_or_event_occurrence = date_time_- select date_time_select = date date => calendar_date
planned_date_effectivity to date (as start_date)	PATH			dated_effectivity.effectivity_start_date -> date_time_or_event_occurrence = date_time_- select date_time_select = date date => calendar_date
planned_date_effectivity to date_and_time (as end_date)	PATH			dated_effectivity dated_effectivity.effectivity_end_date -> date_and_time
planned_date_effectivity to date_and_time (as start_date)	PATH			dated_effectivity dated_effectivity.effectivity_start_date -> date_and_time
PLANNED_EFFECTIVITY	configuration_effectivity	44	248, 249	
planned_effectivity to assembly_ composition_relationship (as effective_assembly)	PATH		339, 354	configuration_effectivity <= product_definition_effectivity product_definition_effectivity.usage -> {product_definition_relationship product_definition_relationship.name = 'assembly_composition'}

Table 85 – Mapping table for planned_effectivity UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship => product_definition_usage => assembly_component_usage
planned_effectivity to data_element (as component_quantity)	PATH			configuration_effectivity <= product_definition_effectivity product_definition_effectivity.usage -> product_definition_relationship => product_definition_usage => assembly_component_usage => quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit
planned_effectivity to ee_approval (as effectivity_approval)	PATH		13	configuration_effectivity approval_assigned_item = configuration_ effectivity approval_assigned_item <= applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
planned_effectivity to ee_product_configuration (as configuration)	PATH			configuration_effectivity configuration_effectivity.configuration -> configuration_design configuration_design.configuration -> configuration_item
PLANNED_LOT_EFFECTIVITY	lot_effectivity	41		
lot_number	lot_effectivity.effectivity_lot_id	41		
planned_lot_effectivity to data_element (as lot_size)	PATH			lot_effectivity lot_effectivity.effectivity_lot_size -> measure_with_unit

Table 85 – Mapping table for planned_effectivity UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
PLANNED_SERIAL_NUM-BER_EFFECTIVITY	serial_numbered_effectivity	41		
from_effectivity_id	serial_numbered_effectivity.start_id	41		
through_effectivity_id	serial_numbered_effectivity.end_id	41		

Table 86 – Mapping table for printed_part_template UoF

Application element	AIM element	Source	Rules	Reference path
NON_CONDUCTIVE_CROSS_SECTION_TEMPLATE	part_template_definition	210		<pre> part_template_definition <= {shape_aspect shape_aspect.description = 'non conductive cross section template'} </pre>
non_conductive_cross_section_- template to length_data_element (as nominal_width)	PATH			<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'nominal width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
non_conductive_cross_section_- template to stratum_technology (as template_technology)	PATH			<pre> part_template_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition <- property_definition_relationship.related_- property_definition </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_relationship { property_definition_relationship property_definition_relationship.name = 'technology usage' } property_definition_relationship.relati- ng_ property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => stratum_technology </pre>
PHYSICAL_COMPONENT_- TERMINAL	(component_terminal) (component_interface_terminal)	210 210		<pre> (component_terminal <= shape_aspect { shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect) </pre>
PRINTED_CONNECTOR_TEM- PLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= { shape_aspect shape_aspect.description = 'printed connector template' } </pre>
PRINTED_CONNECTOR_TEM- PLATE_TERMINAL_RELA- TIONSHIP	printed_connector_template_- terminal_relationship	210	257	<pre> printed_connector_template_terminal_- relationship <= [shape_aspect [shape_aspect_relationship] printed_connector_template_terminal_- relationship <= shape_aspect <- shape_aspect_relationship.related_shape_aspect { shape_aspect_relationship shape_aspect_relationship.name = 'connector' } shape_aspect_relationship.relati- ng_shape_- aspect -> </pre>
printed_connector_template_- terminal_relationship to printed_connector_template (as connector)	PATH		340	

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect shape_aspect.description = 'printed connector template'} shape_aspect => part_template_definition
printed_connector_template_- terminal_relationship to printed_part_template_interface_- terminal (as interface_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'interface terminal'} shape_aspect => printed_part_template_terminal
printed_connector_template_- terminal_relationship to printed_part_template_join_- terminal (as join_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'join terminal'} shape_aspect => printed_part_template_terminal
PRINTED_PART_CROSS_SECTION_TEMPLATE	part_template_definition	210	333, 340	part_template_definition <= {shape_aspect shape_aspect.description = 'printed part cross section template'}
printed_part_cross_section_- template to analytical_model (as transmission_line_model)	PATH		333, 340	part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation ->

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_cross_section_- template to analytical_representation (as maximum_transmission_line_- characteristic)	PATH			<pre> representation => analytical_model part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'maximum transmission line characteristic'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation property_definition_representation.used_- representation -> representation => analytical_representation </pre>
printed_part_cross_section_- template to analytical_representation (as minimum_transmission_line_- characteristic)	PATH			<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'minimum transmission line characteristic'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation property_definition_representation.used_- representation -> representation => analytical_representation </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_cross_section_- template to edge_segment_vertex (as end_vertex)	PATH			<pre> part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'end vertex'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
printed_part_cross_section_- template to edge_segment_vertex (as start_vertex)	PATH			<pre> part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'start vertex'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.name = 'single datum'} shape_aspect => physical_unit_datum => edge_segment_vertex </pre>
printed_part_cross_section_- template to length_data_element (as maximum_width)	PATH			<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item.name = 'maximum width' representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
printed_part_cross_section_- template to length_data_element (as minimum_width)	PATH		333, 340	<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item.name = 'minimum width' representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_cross_section_- template to printed_part_template_material_- link (as horizontal_material_link)	PATH			<pre> part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'horizontal material link'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>
printed_part_cross_section_- template to printed_part_template_material_- link (as vertical_material_link)	PATH			<pre> part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'vertical material link'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>
PRINTED_PART_CROSS_SECTION_CROSS_SECTION_TERMINAL_TEMPLATE_TERMINAL	printed_part_cross_section_- template_terminal	210		<pre> printed_part_cross_section_template_terminal <= {[printed_part_template_terminal group_assigned_item = printed_part_template_- terminal group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> group_assignment group_assignment.assigned_group -> {group => printed_part_template_terminal_connection_- zone_category} group group.name {(group.name = 'area edge segment')} (group.name = 'curve edge segment')}}] [printed_part_template_terminal <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition [product_definition product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model'] [product_definition.frame_of_reference -> product_definition_context <= </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> application_context_element application_context_element.name = 'template definition']]] </pre>
<pre> printed_part_cross_section_- template_terminal to printed_part_cross_section_- template (as cross_section_definition) </pre>	PATH			<pre> printed_part_cross_section_template_terminal <= printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'cross section_definition'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'printed part cross section template'} shape_aspect => part_template_definition printed_part_cross_section_template_terminal <= printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material to left of terminal'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>
<pre> printed_part_cross_section_- template_terminal to printed_part_template_material_- link (as material_to_left_of_terminal) </pre>	PATH			<pre> printed_part_cross_section_template_terminal <= printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material to left of terminal'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_cross_section_- template_terminal to printed_part_template_material_- link (as material_to_right_of_terminal)	PATH			<pre> printed_part_cross_section_template_terminal <= printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material to right of terminal'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => printed_part_template_material_link printed_part_cross_section_template_terminal </pre>
printed_part_cross_section_- template_terminal to printed_part_template_material_- link (as material_to_top_of_terminal)	PATH			<pre> printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material to top of terminal'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_cross_section_terminal - template_terminal to printed_part_template_material_- link (as material_to_bottom_of_terminal)	PATH			<pre> printed_part_cross_section_template_terminal <= printed_part_template_terminal <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material to bottom of terminal'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => printed_part_template_material_link </pre>
PRINTED_PART_TEMPLATE	part_template_definition	210	333, 340	<pre> part_template_definition <= {shape_aspect shape_aspect.description = 'printed part template'} </pre>
printed_part_template to functional_unit_usage_view (as implemented_function)	PATH		333, 340, 338	<pre> part_template_definition <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition <- property_definition_relationship.related_- property_definition property_definition_relationship {property_definition_relationship.name = 'implemented function'} property_definition_relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional design usage'} product_definition => (functional_unit) (functional_unit => externally_defined_functional_unit) (functional_unit => externally_defined_functional_unit => library_defined_functional_unit) part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'printed part template stack'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition => stratum_technology_link => printed_part_template_link part_template_definition <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship </pre>
printed_part_template to printed_part_template_link (as printed_part_template_stack)	PATH			
printed_part_template to stratum_technology (as of_stratum_technology)	PATH			

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect_relationship.name = 'of stratum technology' shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_ object characterized_object => stratum_technology </pre>
printed_part_template to stratum_technology_link (as material_stack)	PATH			<pre> part_template_definition <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_ property_definition property_definition.relationship {property_definition_relationship property_definition_relationship.name = 'material_stack'} property_definition.relationship.relatng_ property_definition -> property_definition => stratum_technology_link </pre>
PRINTED_PART_TEMPLATE_- CONNECTED_TERMINALS_- DEFINITION	printed_part_template_- connected_terminals_definition	210		printed_part_template_connected_terminals_- definition <=
name	shape_aspect.name	41	333	<pre> shape_aspect </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_template_- connected_terminals_definition to printed_part_template_terminal (as connected_terminals)	PATH		333	printed_part_template_connected_terminals_- definition <= shape_aspect <= shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.description = 'member_connected_terminal'} shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join_terminal')} shape_aspect => printed_part_template_terminal
PRINTED_PART_TEMPLATE_- INTERFACE_TERMINAL	printed_part_template_terminal	210		printed_part_template_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interface terminal'}
printed_part_template_interface_- terminal to printed_connector_template (as associated_definition)	PATH			printed_part_template_terminal <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated definition'} shape_aspect_relationship.relatng_shape_- aspect ->

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_aspect => {shape_aspect shape_aspect.description = 'printed connector template'} part_template_definition </pre>
PRINTED_PART_TEMPLATE_- JOIN_TERMINAL	printed_part_template_terminal	210	333	<pre> printed_part_template_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'join terminal'} </pre>
PRINTED_PART_TEMPLATE_- LINK	printed_part_template_link	210		<pre> {[printed_part_template_link <= stratum_technology_link <= property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => {shape_aspect shape_aspect.description = 'printed part template'} part_template_definition] [printed_part_template_link <= shape_aspect_relationship]} </pre>
printed_part_template_link to printed_part_template (as precedent_printed_part_template)	PATH			<pre> printed_part_template_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => part_template_definition </pre>
printed_part_template_link to printed_part_template (as subsequent_printed_part_- template)	PATH			<pre> printed_part_template_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => part_template_definition </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PRINTED_PART_TEMPLATE_- MATERIAL	printed_part_template_material	210		printed_part_template_material <= product_definition
printed_part_template_material to length_data_element (as maximum_width)	PATH			printed_part_template_material <= product_definition characterized_product_definition = product_- definition characterized_product_definition product_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
printed_part_template_material to length_data_element (as minimum_width)	PATH			printed_part_template_material <= product_definition characterized_product_definition = product_- definition characterized_product_definition product_definition = characterized_- product_definition characterized_definition <=

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
printed_part_template_material to stratum_technology (as required_material)	PATH			<pre> printed_part_template_material <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship property_definition_relationship.related_ property_definition -> </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => stratum_technology
PRINTED_PART_TEMPLATE_- MATERIAL_LINK	printed_part_template_material_- link	210		{{[printed_part_template_material_link <= shape_aspect] [printed_part_template_material_link <= shape_aspect_relationship]}}
printed_part_template_material_- link to template_material_cross_section_- boundary (as precedent_material)	PATH			printed_part_template_material_link <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => template_material_cross_section_boundary
printed_part_template_material_- link to template_material_cross_section_- boundary (as subsequent_material)	PATH			printed_part_template_material_link <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => template_material_cross_section_boundary
PRINTED_PART_TEMPLATE_- TERMINAL	printed_part_template_terminal	210	333, 259	printed_part_template_terminal <= shape_aspect {{[shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')] [shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'template model' [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'template definition'[]]}
connection_zone_category	group.name	41		printed_part_template_terminal group_assigned_item = printed_part_template_ terminal group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> {group => printed_part_template_terminal_connection_ zone_category} group group.name {(group.name = 'area edge segment')}

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(group.name = 'curve edge segment') (group.name = 'surface area') (group.name = 'surface point')}
name	shape_aspect.name	41	333	printed_part_template_terminal <= shape_aspect shape_aspect.name
printed_part_template_terminal to connection_zone (as terminal_connection_zone)	PATH		333	printed_part_template_terminal <= shape_aspect <= shape_aspect.relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'terminal connection zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
printed_part_template_terminal to printed_part_template (as associated_definition)	PATH		333, 340	printed_part_template_terminal <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'associated definition'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect shape_aspect.description = 'printed part template'} shape_aspect => part_template_definition
PRINTED_PART_TEMPLATE_- TERMINAL_RELATIONSHIP	shape_aspect_relationship	41	258	{shape_aspect_relationship shape_aspect_relationship.name = 'printed part template terminal relationship'}

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
printed_part_template_terminal_- relationship to printed_part_template_terminal (as related_printed_part_terminal)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')}} shape_aspect => printed_part_template_terminal
printed_part_template_terminal_- relationship to printed_part_template_terminal (as relating_printed_part_terminal)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')}} shape_aspect => printed_part_template_terminal
STRATUM_TECHNOLOGY_- LINK	stratum_technology_link	210		{{[stratum_technology_link <= property_definition_relationship] [stratum_technology_link <= property_definition]]}
stratum_technology_link to stratum_technology (as precedent_stratum_technology)	PATH			stratum_technology_link <= property_definition_relationship property_definition_relationship.relatng_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object characterized_object => stratum_technology
stratum_technology_link to stratum_technology (as subsequent_stratum_technology)	PATH			stratum_technology_link <= property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition ->

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_ object characterized_object => stratum_technology
stratum_technology_link to stratum_technology_link (as precedent_stratum_ technology_link)	PATH			stratum_technology_link <= property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'precedent_stratum_technology link'} property_definition_relationship.related_ property_definition -> property_definition => stratum_technology_link
TEMPLATE_MATERIAL_ CROSS_SECTION_BOUNDARY	template_material_cross_section_ boundary	210		template_material_cross_section_boundary <= shape_aspect
name	PATH			template_material_cross_section_boundary <= shape_aspect {shape_aspect (shape_aspect.name = 'top') (shape_aspect.name = 'left') (shape_aspect.name = 'bottom') (shape_aspect.name = 'right')}} shape_aspect.name
template_material_cross_section_ boundary to printed_part_template_material (as associated_definition)	PATH			template_material_cross_section_boundary <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => printed_part_template_material
USAGE_CONCEPT_USAGE_- RELATIONSHIP	mapped_item	43		{mapped_item <= mapped_item <= representation_item representation_item.name = 'ucur'} mapped_item.mapping_source -> representation_map representation_map.mapping_origin -> representation_item => geometric_representation_item => placement}
usage_concept_usage_relationship to axis_placement (as associated_usage_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement
usage_concept_usage_relationship to cartesian_transformation_ operator_2d (as associated_usage_placement)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => cartesian_transformation_operator_2d
usage_concept_usage_relationship to component_termination_ passage_template_terminal (as associated_usage)	PATH			mapped_item <= representation_item <= representation.items[i] {representation => shape_representation} representation <= property_definition_representation.used_ representation

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect (shape_aspect.description = 'component termination passage template interface terminal') (shape_aspect.description = 'component termination passage template join terminal')} mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation </pre>
usage_concept_usage_relationship to shape_definition (as associated_shape_definition)	PATH			
usage_concept_usage_relationship to land_template_terminal (as associated_usage)	PATH		333	<pre> mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect (shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal') shape_aspect => land_template_terminal
usage_concept_usage_relationship to non_feature_shape_element (as associated_usage)	PATH			mapped_item <= representation_item <- representation.items[i] {representation => shape_representation} representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}
usage_concept_usage_relationship to non_feature_shape_definition (as associating_non_feature_ usage_shape)	PATH			mapped_item <= representation_item <- representation.items[i] representation {representation => shape_representation} representation <-

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}}</pre>
usage_concept_usage_relationship to part_feature (as associated_usage)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature'])}</pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'part generic feature') (shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect =>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal => bare_die_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal) (shape_aspect => package_body_surface)) </pre>
usage_concept_usage_relationship to part_template_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation </pre>
usage_concept_usage_relationship to part_template_non_planar_2d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'part template non planar 2d shape']} [representation <- </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation => manifold_surface_shape_representation </pre>
usage_concept_usage_relationship to part_template_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition[] representation => shape_representation </pre>
usage_concept_usage_relationship to physical_unit_3d_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] representation => shape_representation } </pre>
usage_concept_usage_relationship to physical_unit_planar_shape (as associating_usage_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] representation {[representation representation.name = 'planar projected shape'] } </pre>

Table 86 – Mapping table for printed_part_template UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit}] representation => shape_representation</pre>
usage_concept_usage_relationship to printed_part_template_terminal (as associated_usage)	PATH		333	<pre>mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect {shape_aspect</pre>

Table 86 – Mapping table for printed_part_template UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interface terminal') (shape_aspect.description = 'join terminal')} shape_aspect => printed_part_template_terminal
usage_concept_usage_relationship to via_template_terminal (as associated_usage)	PATH			mapped_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'via template terminal'}

Table 87 – Mapping table for product_identification UoF

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT	product	41	155, 159	
description	product.description	41		
product_nomenclature	product.name	41		
part_number	product.id	41		
standard_product_indicator	product_category.name	41		product <- product_related_product_category.products[i] product_related_product_category <= product_category <- product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category {product_category product_category.name = 'standard part'} product_category.name
ee_product to organization (as design_owner)	PATH		106	product organization_assigned_item = product organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'design owner'} organization_assignment organization_assignment.assigned_organization -> organization

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product to person_and_organization (as design_owner)	PATH		108	product person_and_organization_assigned_item = product person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'design owner'} person_and_organization_assignment person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization
EE_PRODUCT_CATEGORY	characterized_product_category	210		characterized_product_category characterized_product_category <= [product_category] [characterized_object]
description	product_category.description	41		characterized_product_category <= product_category product_category.description
name	product_category.name	41		characterized_product_category <= product_category product_category.name
ee_product_category to model_parameter (as category_parameter)	PATH		340, 109	characterized_product_category <= characterized_object characterized_definition = characterized_- object characterized_definition <- characterized_definition <-

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition.definition property_definition {property_definition.description = 'product category parameter'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => model_parameter </pre>
ee_product_category to parameter_assignment (as category_specific_values)	PATH			<pre> characterized_product_category <= characterized_object characterized_definition = characterized_ object characterized_definition characterized_definition <- property_definition.definition property_definition {property_definition.description = 'product category values'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation => parameter_assignment_representation </pre>
ee_product_category to ee_document (as reference_document)	PATH			<pre> characterized_product_category <= characterized_object document_assigned_item = characterized_object document_assigned_item <- </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				applied_document_reference.items[i] applied_document_reference document_reference {document_reference.role -> object_role object_role.name = 'product category reference document'} document_reference.assigned_document -> (document) (document ==> ee_specification)
EE_PRODUCT_CATEGORY_RELATIONSHIP	product_category_relationship	41		product_category_relationship
ee_product_category_relationship to ee_product_category (as category)	PATH			product_category_relationship product_category_relationship.category -> product_category => characterized product category
ee_product_category_relationship to ee_product_category (as sub_category)	PATH			product_category_relationship product_category_relationship.sub_category -> product_category => characterized product category
EE_PRODUCT_DEFINITION	(product_definition_with_ - associated_documents) (product_definition) (evaluation_product_definition) (physical_unit) (rule_definition) (externally_defined_product_ - definition) (externally_defined_physical_unit) (library_defined_physical_unit) (analytical_model_definition) (footprint_definition) (padstack_definition) (physical_unit_network_ - definition) (requirement_definition) (stratum) (functional_unit) (externally_defined_functional_ - unit) (library_defined_functional_unit)	41 41 210 210 210 210 210 210 210 210 210 210 210 210 210 210 210 210	156, 157	(product_definition_with_associated_documents <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition'))} (evaluation_product_definition <= product_definition) (rule_definition <= product_definition) (physical_unit <= product_definition) ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (library_defined_physical_unit <= externally_defined_physical_unit <= [externally_defined_product_definition])

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [physical_unit <= product_definition]) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (functional_unit <= product_definition) (externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) (library_defined_functional_unit <= externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition]) </pre>
cad_filename	document.id	41	338	<pre> product_definition_with_associated_documents product_definition_with_associated_ - documents.documentation_ids[i] -> document {document.kind -> document_type document_type.product_data_type = 'cad filename'} document.id </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
life_cycle_status	product_definition_context.life_cycle_stage	41		<pre> ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <=) (physical_unit <=) (library_defined_physical_unit <=) (externally_defined_physical_unit <=) (physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <=) (functional_unit <=) (library_defined_functional_unit <=) (externally_defined_functional_unit <=) (functional_unit <=) (product_definition) ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition) (product_definition) </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'template definition')} (application_context_element.name = 'material definition')} product_definition.frame_of_reference -> product_definition_context product_definition_context.life_cycle_stage product_definition_with_associated_documents <= (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) </pre>
product_definition_id	product_definition.id	41	338	

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (product_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}} product_definition.id ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition) </pre>
ee_product_definition to date_and_time (as creation_date)	PATH		265, 338, 72	<pre> (product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition) </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}) date_and_time_assigned_item = product_ definition date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment.role -> </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				date_time_role date_time_role.name = 'creation date' date_and_time_assignment.assigned_date_and_time -> date and time
ee_product_definition_to_ee_approval (as product_definition_approval)	PATH		13	((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) (product_definition) ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <=

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}) approval_assigned_item = product_definition approval_assigned_item <- applied_approval_assignment.items[i] approval_assignment <= approval_assignment.applied_approval -> approval </pre>
ee_product_definition to ee_document (as documentation)	PATH		338	<pre> ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				functional_unit <= product_definition ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}) document_assigned_item = product_definition document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				document_reference.assigned_document -> (document) (document => ee_specification)
ee_product_definition to ee_product_version (as version)	PATH			((product_definition_with_associated_- documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition)

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}) product_definition.format -> product_definition.format {product_definition.format.description != rule version'}) (rule_definition <= product_definition product_definition.format -> product_definition.format {product_definition.format.description = rule version'}) </pre>
ee_product_definition to ee_text (as context_description)	PATH			<pre> ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition')}) (application_context_element.name = 'material definition')}) characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition {property_definition property_definition.name = 'product definition context description'} property_definition <- property_definition.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => descriptive_representation_item {descriptive_representation_item descriptive_representation_item.name = 'product definition context description'} ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition)
ee_product_definition to ee_text (as discipline_id)	PATH			

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'template definition') (application_context_element.name = 'material definition')}) characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition { property_definition property_definition.name = 'product definition discipline id' } property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => descriptive_representation_item { descriptive_representation_item descriptive_representation_item.name = 'product definition discipline id' } </pre>
ee_product_definition to organization (as creator)	PATH		266, 338	<pre> ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition) ([externally_defined_product_definition <= product_definition]) </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'template definition')}) (application_context_element.name = 'material definition')}) organization_assigned_item = product_ definition organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment organization_assignment.role -> organization_role </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> organization_role.name = 'creator' } organization_assignment.assigned_organization -> organization </pre>
ee_product_definition_to_person_and_organization (as creator)	PATH			<pre> ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <= externally_defined_physical_unit <= physical_unit <=) (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition) ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (requirement_definition <= product_definition) (stratum <= product_definition) {product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'template definition')} (application_context_element.name = 'material definition')} person_and_organization_assigned_item = product_definition person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'creator'} person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization } ((product_definition_with_associated_documents <=) (physical_unit <=) (externally_defined_physical_unit <= physical_unit <=) (library_defined_physical_unit <=) </pre>
ee_product_definition to security_classification (as security_code)	PATH		338	

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit <= physical_unit <= (functional_unit <=) (externally_defined_functional_unit <= functional_unit <=) (library_defined_functional_unit <= externally_defined_functional_unit <= functional_unit <=) product_definition ([externally_defined_product_definition <= product_definition] [externally_defined_product_definition <= externally_defined_item]) (rule_definition <= product_definition) (analytical_model_definition <= product_definition) (footprint_definition <= product_definition) (padstack_definition <= product_definition) (physical_unit_network_definition <= product_definition) (requirement_definition <= product_definition) (stratum <= product_definition) (product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'template definition'})

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(application_context_element.name = 'material_definition')) security_classification_assigned_item = product_definition security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification</pre>
EE_PRODUCT_RELATED_- PRODUCT_CATEGORY	product_related_characterized_- product_category	210		<pre>product_related_characterized_product_category <= characterized_product_category <= product_category {product_category => product_related_product_category}</pre>
ee_product_related_product_- category to ee_product (as of_product)	PATH			<pre>product_related_characterized_product_category <= product_related_product_category product_related_product_category.products[i] -> product</pre>
EE_PRODUCT_SPECIFIC_PA- RAMETER_VALUE_ASSIGN- MENT	product_specific_parameter_- value_assignment	210		<pre>product_specific_parameter_value_assignment <= product_related_characterized_product_category <= characterized_product_category <= product_category {product_category => product_related_product_category}</pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product_specific_parameter_value_assignment to parameter_assignment (as assigned_parameter)	PATH		340, 109	<pre> product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category <= product_category => characterized_product_category <= characterized_object characterized_definition = characterized_ object characterized_definition characterized_definition <- property_definition.definition property_definition {property_definition.description = 'product category parameter'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item => (model_parameter) (measure_representation_item) (descriptive_representation_item) </pre>
ee_product_specific_parameter_value_assignment to ee_product_related_product_category (as associated_product_category)	PATH			<pre> product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category <= product_category <- product_category_relationship.sub_category product_category_relationship {product_category_relationship.description = 'product value assignment'} product_category_relationship.category -> </pre>

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_category => product_related_product_category => product_related_characterized_product_category <= product_specific_parameter_value_assignment <= product_related_characterized_product_category <= product_related_product_category product_related_product_category.products[i] -> product
EE_PRODUCT_VERSION		41		
revision_code	product_definition_formation	41		
ee_product_version to ee_approval (as product_version_approval)	PATH			product_definition_formation approval_assigned_item = product_definition_formation approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
ee_product_version to contract (as authorizing_agreement)	PATH		263	product_definition_formation contract_assigned_item = product_definition_formation contract_assigned_item <- applied_contract_assignment.items[i] applied_contract_assignment <= contract_assignment contract_assignment.assigned_contract -> contract
ee_product_version to ee_product (as versioned_item)	PATH		268	product_definition_formation product_definition_formation.of_product -> product

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product_version to organization (as creator)	PATH		106	product_definition_formation organization_assigned_item = product_ definition_formation organization_assigned_item <- applied_organization_assignment.items[1] applied_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'creator'} organization_assignment organization_assignment.assigned_organization -> organization
ee_product_version to person_and_organization (as creator)	PATH		108	product_definition_formation person_and_organization_assigned_item = product_definition_formation person_and_organization_assigned_item <- applied_person_and_organization_ assignment.items[1] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'creator'} person_and_organization_assignment person_and_organization_assignment.assigned_ person_and_organization -> person_and_organization
ee_product_version to security_classification (as security_code)	PATH			product_definition_formation security_classification_assigned_item = product_definition_formation security_classification_assigned_item <- applied_security_classification_ assignment.items[1] applied_security_classification_assignment <=

Table 87 – Mapping table for product_identification UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification
EE_PRODUCT_VERSION_SU- PERSEDECE	product_definition_formation_ relationship	41		{product_definition_formation_relationship product_definition_formation_relationship.name = 'product version supersedece'}
product_association to ee_product_version (as superseded_product_version)	PATH			product_definition_formation_relationship product_definition_formation_ relationship.related_product_definition_ formation -> product_definition_formation
product_association to ee_product_version (as superseding_product_version)	PATH			product_definition_formation_relationship product_definition_formation_ relationship.relatng_product_definition_ formation -> product_definition_formation
EXTERNALLY_DEFINED_ EE_PRODUCT_DEFINITION	(externally_defined_product_ definition) (externally_defined_physical_unit) (externally_defined_functional_ unit)	210 210 210		(externally_defined_product_definition <= [product_definition] [externally_defined_item]) (externally_defined_physical_unit <= [externally_defined_product_definition] [physical_unit <= product_definition]) (externally_defined_functional_unit <= [externally_defined_product_definition] [functional_unit <= product_definition])
source	external_source.source_id	41	351	(externally_defined_product_definition <= (externally_defined_physical_unit <= (externally_defined_functional_unit <= externally_defined_item external_source -> external_source.source_id

Table 87 – Mapping table for product_identification UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_ASSOCIATION	product_definition_formation_ - relationship	41	260, 261	{product_definition_formation_relationship product_definition_formation_relationship.name = 'product association'}
product_association to ee_product_version (as related_product)	PATH			product_definition_formation_relationship product_definition_formation_ - relationship.related_product_definition_ - formation ->
product_association to ee_product_version (as relating_product)	PATH			product_definition_formation product_definition_formation_relationship product_definition_formation_ - relationship.relatng_product_definition_ - formation ->
product_association to ee_specification (as constraint_source)	PATH			product_definition_formation product_definition_formation_relationship document_assigned_item = product_definition_ - formation_relationship document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document => ee_specification

Table 88 – Mapping table for product_requirement_allocation UoF

Application element	AIM element	Source	Rules	Reference path
REQUIREMENT_ALLOCATION	(product_definition_relationship) (shape_aspect_relationship) (requirement_allocation_group)	41 41 210		<pre> ({product_definition_relationship product_definition_relationship.name = 'requirement_allocation'}) ({shape_aspect_relationship shape_aspect_relationship.name = 'requirement_allocation'}) ({[requirement_allocation_group <= property_definition_relationship] [requirement_allocation_group <= group]}) requirement_allocation_group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment.items[i] -> group_assigned_item (group_assigned_item = product) (group_assigned_item = product_definition_ formation) (group_assigned_item = configuration_item)}) requirement_allocation_group <= requirement_allocation_group group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment.items[i] -> group_assigned_item = representation {representation.name = 'component external reference'}} </pre>
requirement_allocation to component_external_reference (as target_product_object)	PATH			<pre> requirement_allocation_group <= requirement_allocation_group group <- group_assignment.assigned_group group_assignment => applied_group_assignment {applied_group_assignment.items[i] -> group_assigned_item = representation {representation.name = 'component external reference'}} </pre>

Table 88 – Mapping table for product_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
requirement_allocation to component_feature_external_- reference (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = representation_item {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item </pre>
requirement_allocation to ee_product (as target_product_object)	PATH		268	<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product </pre>
requirement_allocation to ee_product_configuration (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = configuration_item </pre>
requirement_allocation to ee_product_version (as target_product_object)	PATH			<pre> requirement_allocation_group requirement_allocation_group <= group <- group_assignment.assigned_group </pre>

Table 88 – Mapping table for product_requirement_allocation UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item = product_definition_- formation
requirement_allocation to ee_requirement_occurrence (as allocated_requirement)	PATH			<pre> (product_definition.relationship product_definition.relationship.relatin_ product_definition -> product_definition {product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => (shape_aspect.relationship shape_aspect.relationship.relatin_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition </pre>

Table 88 – Mapping table for product_requirement_allocation UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition => (requirement_allocation_group <= property_definition_relationship property_definition_relationship.relatin_- property_definition -> property_definition =>) requirements_property </pre>
requirement_allocation to ee_requirement_occurrence (as target_product_object)	PATH			<pre> requirement_allocation_group <= property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'derived from operation'} property_definition_relationship.related_- property_definition -> property_definition => requirements_property </pre>

Table 89 – Mapping table for product_rule UoF

Application element	AIM element	Source	Rules	Reference path
EE_PRODUCT	product	41	155, 159	
description	product.description	41		
product_nomenclature	product.name	41		
part_number	product.id	41		
standard_product_indicator	product_category.name	41		product <- product_related_product_category.products[i] product_related_product_category <= product_category <- product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category {product_category product_category.name = 'standard part'} product_category.name
ee_product to organization (as design_owner)	PATH		106	product organization_assigned_item = product organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'design owner'} organization_assignment organization_assignment.assigned_organization -> organization

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product to person_and_organization (as design_owner)	PATH		108	product person_and_organization_assigned_item = product person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'design owner'} person_and_organization_assignment person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization
EE_RULE_COMPLEX_CLAUSE	rule_complex_clause	210		rule_complex_clause <= representation
identifier	representation.name	43		rule_complex_clause <= representation representation.name
ee_rule_complex_clause to ee_rule_simple_clause (as conjunctive_content)	PATH			rule_complex_clause <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => rule_simple_clause
EE_RULE_CONCLUSION_DEFINITION	rule_conclusion_definition	210	161	rule_conclusion_definition <= representation
identifier	representation.name	43		rule_conclusion_definition <= representation representation.name
ee_rule_conclusion_definition to ee_rule_definition (as rule_definition)	PATH			rule_conclusion_definition <= representation <- property_definition_representation.used_- representation property_definition_representation.definition -> property_definition.definition -> characterized_definition = characterized_- product_definition

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition product_definition => rule_definition
ee_rule_conclusion_definition to ee_rule_simple_clause (as conjunctive_content)	PATH			rule_conclusion_definition <= representation <- representation_relationship.rep_2 representation_relationship.rep_1 -> representation => rule_simple_clause
EE_RULE_DEFINITION	rule_definition	210		{rule_definition <= product_definition} rule_definition <= product_definition
ee_rule_definition to ee_rule_version (as version)	PATH			product_definition.formation -> product_definition_formation {product_definition_formation.description = 'rule version'}
EE_RULE_FUNCTION	rule_function_definition	210	342, 162	rule_function_definition <= representation
identifier	representation.name	43	342	rule_function_definition <= representation.name
ee_rule_function to model_parameter (as parameter)	PATH			rule_function_definition <= representation.items[i] -> representation_item => model_parameter
ee_rule_function to ee_document (as reference_document)	PATH			rule_function_definition <= representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i]

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code> applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'reference document'} (document) (document => ee_specification) </code>
ee_rule_function to ee_specification (as source_code)	PATH			<code> rule_function_definition <= representation document_assigned_item = representation document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'source code'} document => ee_specification </code>
EE_RULE_PREMISE_DEFINITION identifier	rule_premise_definition representation.name	210 43	163	<code> rule_premise_definition <= representation rule_premise_definition <= representation representation.name rule_premise_definition <= representation <- property_definition_representation.used_ - representation property_definition_representation.definition -> property_definition.definition -> characterized_definition = characterized_ - product_definition </code>

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition product_definition => rule_definition
ee_rule_premise_definition to ee_rule_complex_clause (as disjunctive_content)	PATH			rule_premise_definition <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => rule_complex_clause
EE_RULE_SIMPLE_CLAUSE	rule_simple_clause	210		rule_simple_clause <= representation
identifier	representation.name	43		rule_simple_clause <= representation representation.name
ee_rule_simple_clause to ee_rule_function (as rule_function)	PATH			rule_simple_clause <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => rule_function_definition
ee_rule_simple_clause to parameter_assignment (as parameter)	PATH			rule_simple_clause <= representation <- representation_relationship.rep_1 representation_relationship representation_relationship.rep_2 -> representation => parameter_assignment_representation
EE_RULE_VERSION	product_definition_formation	41		{product_definition_formation product_definition_formation.description = 'rule version'}
PARTICIPANT_DATE_AND_- TIME_ASSIGNMENT	applied_date_and_time_- assignment	210		applied_date_and_time_assignment

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
participant_date_and_time_- assignment to date_and_time (as action_date_and_time)	PATH		103	<pre> applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'participant date and time'} date_and_time_assignment.assigned_date_and_- time -> date_and_time </pre>
participant_date_and_time_- assignment to person_and_organization (as action_participant)	PATH			<pre> applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_assigned_item date_and_time_assigned_item = rule_action rule_action person_and_organization_assigned_item = rule_- action person_and_organization_assigned_item <- applied_person_and_organization_- assignment.items[i] applied_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.assigned_- person_and_organization -> person_and_organization </pre>
RULE_ACTION	rule_action	210	345, 291	<pre> rule_action <= action </pre>
rule_action to ee_rule_version (as subject_rule)	PATH		345, 332	<pre> rule_action <= action <- action_assignment.assigned_action action_assignment => applied_action_assignment applied_action_assignment.items[i] -> action_assigned_item action_assigned_item = product_definition_- formation {product_definition_formation.description = 'rule version'} </pre>

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
rule_action to participant_date_- and_time_assignment (as subject_action_assignment)	PATH			rule_action date_and_time_assigned_item = rule_action date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment
RULE_ADOPTION	rule_action	210	345	rule_action <= action {action action.name = 'rule adoption'}
RULE_CHANGE_REQUEST	rule_action	210		rule_action <= action {action action.name = 'rule change request'}
change_reason	action.description	41	345	rule_action <= action action.description
RULE_CREATION	rule_action	210	345	rule_action <= action {action action.name = 'rule creation'}
RULE_EXPIRATION	rule_action	210		rule_action <= action {action action.name = 'rule expiration'}
expiration_rationale	action.description	41	345	rule_action <= action action.description
RULE_JUSTIFICATION	rule_action	210	345	rule_action <= action {action action.name = 'rule justification'}
justification_rationale	action.description	41	345	rule_action <= action action.description
rule_justification to rule_action (as justified_action)	PATH		345	rule_action <= action_relationship.related_action {action_relationship

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> action_relationship.name = 'justified action' action_relationship action_relationship.relation_action -> action => rule action </pre>
RULE_MODIFICATION	rule_action	210		<pre> rule_action <= action {action action.name = 'rule modification'} </pre>
rule_modification to rule_change_request (as modification_rationale)	PATH		345	<pre> rule_action <= action <- action_relationship.related_action {action_relationship action_relationship.name = 'modification rationale'} action_relationship action_relationship.relation_action -> {action action.name = 'rule change request'} action => rule_action </pre>
RULE_PRIORITY	representation	43		<pre> {representation representation.name = 'rule priority'} </pre>
priority	descriptive_representation_item	45		<pre> representation representation.items[i] -> representation_item => descriptive_representation_item </pre>
rule_priority to ee_rule_definition (as prioritized_rule)	PATH			<pre> representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition </pre>

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition => rule_definition
RULE_REJECTION	rule_action	210	345	rule_action <= action {action action.name = 'rule rejection'}
rejection_reason	action.description	41	345	rule_action <= action action.description
RULE_REQUEST	rule_action	210	345	rule_action <= action {action action.name = 'rule request'}
RULE_SET	rule_set	210		rule_set <= group
identifier	group.name	41		rule_set <= group group.name
purpose	group.description	41		rule_set <= group group.description
rule_set to rule_priority (as rule_priority_member)	PATH			rule_set <= group <- group_assignment.assigned_group group_assignment => applied_group_assignment applied_group_assignment.items[i] -> group_assigned_item group_assigned_item = rule_definition

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<code>rule_definition <=</code> <code>product_definition</code> <code>characterized_product_definition = product_</code> <code>definition</code> <code>characterized_definition = characterized_</code> <code>product_definition</code> <code>characterized_definition <-</code> <code>property_definition.definition</code> <code>property_definition <-</code> <code>property_definition_representation.definition</code> <code>property_definition_representation</code> <code>property_definition_representation.used_</code> <code>representation -></code> <code>representation</code> <code>{representation</code> <code>representation.name = 'rule priority'}</code>
RULE_SET_GROUP	rule_set	210		<code>rule_set <=</code> <code>group</code>
rule_set_group to rule_set (as elements)	PATH			<code>rule_set <=</code> <code>group <-</code> <code>group_relationship.relatng_group</code> <code>{group_relationship</code> <code>group_relationship.name = 'rule set group</code> <code>element'}</code> <code>group_relationship</code> <code>group_relationship.related_group -></code> <code>group =></code> <code>rule_set</code>
RULE_SUBMISSION	rule_action	210	345	<code>rule_action <=</code> <code>action</code> <code>{action</code> <code>action.name = 'rule submission'}</code>
submission_rationale	action.description	41	345	<code>rule_action <=</code> <code>action</code> <code>action.description</code>

Table 89 – Mapping table for product_rule UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
RULE_SUPERSEDE	rule_action	210	292, 345	<pre> rule_action <= action [action action.name = 'rule replacement to'] [action <- action_assignment.assigned_action rule_replacement_to_assignment rule_replacement_to_assignment.items[i] -> replacement_to_item replacement_to_item = product_definition_ formation product_definition_formation {product_definition_formation.description = 'rule version'}]] </pre>
rule_supersede to ee_rule_version (as superseded_rule)	PATH		345	<pre> rule_action <= action {action action.name = 'rule replacement to'} action <- action_relationship.relation_action action_relationship {action_relationship action_relationship.name = 'rule supersede'} action_relationship.related_action -> action {action => rule_action} {action action.name = 'rule replacement from'} action <- action_assignment.assigned_action action_assignment => rule_replacement_from_assignment </pre>

Table 89 – Mapping table for product_rule UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				rule_replacement_from_assignment.items[i] -> replacement_from_item replacement_from_item = product_definition_ formation product_definition_formation {product_definition_formation.description = 'rule version'}

Table 90 – Mapping table for requirement_decomposition UoF

Application element	AIM element	Source	Rules	Reference path
CONSTRAINT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition (property_definition.id = 'rule based constraint')}
DESIGN_CHARACTERISTIC_OCCURRENCE	requirements_property	210	340	((property_definition.description = 'shape and product definition based constraint') (property_definition.description = 'product definition based constraint'))}
DESIGN_REQUIREMENT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition (property_definition.description = 'design characteristic')}
EE_PRODUCT_DEFINITION_BASED_CONSTRAINT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition (property_definition.description = 'shape and product definition based constraint') (property_definition.description = 'product definition based constraint'))}

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
ee_product_definition_based_- constraint_occurrence to ee_product_definition (as constraining_part)	PATH			<pre> requirements_property <= property_definition <- property_definition_relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'constraining_product_definition'} property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition (characterized_product_definition = product_- definition) (characterized_product_definition = product_- definition => product_definition_with_associated_documents) (characterized_product_definition = product_- definition => physical_unit) (characterized_product_definition = product_- definition => externally_defined_physical_unit) (characterized_product_definition = product_- definition => externally_defined_physical_unit => library_defined_physical_unit) (characterized_product_definition = product_- definition => analytical_model_definition) (characterized_product_definition = product_- definition => </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> footprint_definition) (characterized_product_definition = product_ definition => padstack_definition) (characterized_product_definition = product_ definition => physical_unit_network_definition) (characterized_product_definition = product_ definition => requirement_definition) (characterized_product_definition = product_ definition => stratum) (characterized_product_definition = product_ definition => functional_unit) (characterized_product_definition = product_ definition => functional_unit => externally_defined_functional_unit) (characterized_product_definition = product_ definition => functional_unit => externally_defined_functional_unit => library_defined_functional_unit) </pre>
EE_REQUIREMENT_OCCURRENCE	requirements_property	210	340	<pre> requirements_property <= property_definition {property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition {[product_definition </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> product_definition.frame_of_reference -> product_definition.context <= application_context_element (application_context_element.name = 'design requirement') (application_context_element.name = 'requirement') [product_definition product_definition.name = 'requirements model']] </pre>
description	PATH		340, 109	<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'requirements description'} representation_item => descriptive_representation_item </pre>
name	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> {representation_item representation_item.name = 'requirements name'} </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => descriptive_representation_item descriptive_representation_item.description </pre>
ee_requirement_occurrence to analytical_representation (as required_analytical_-representation)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation => analytical_representation </pre>
ee_requirement_occurrence to characteristic (as required_characteristic)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation representation.items[i] -> representation_item {representation_item group_assigned_item = representation_item group_assigned_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment group_assignment.assigned_group -> group => characteristic_type} </pre>
ee_requirement_occurrence to coordinated_characteristic (as required_coordinated_-characteristic)	PATH		340, 109	<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation representation.items[i] -> representation_item => coordinated_representation_item requirements_property <= property_definition property_definition.definition -> characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'requirement'} characterized_product_definition = product_ definition characterized_definition = characterized_ product_definition characterized_definition <- material_designation.definitions[i] material_designation </pre>
ee_requirement_occurrence to ee_material (as required_material)	PATH			
ee_requirement_occurrence to ee_product (as required_part)	PATH		340, 265, 266, 263, 264, 268, 267	<pre> requirements_property <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {[product_definition => requirement_definition] [product_definition product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'requirement'[]] product_definition <- product_definition.relationship.relatiing_- product_definition product_definition.relationship {product_definition.relationship product_definition.relationship.name = 'required parts' } product_definition.relationship.related_- product_definition -> product_definition product_definition.formatiion -> product_definition.formatiion product_definition.formatiion.of_product -> product requirements_property <= property_definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_- representation -> representation => functional_specification requirements_property <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition </pre>
ee_requirement_occurrence to functional_specification (as required_functional_specification)	PATH			
ee_requirement_occurrence to ee_product_definition (as associated_definition)	PATH			

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition product_definition => requirement_definition
ee_requirement_occurrence to ee_specification (as requirement_specification)	PATH			requirements_property document_assigned_item = requirements_property document_assigned_item <= applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document => ee_specification
ELECTRICAL_REQUIRE- MENT_OCCURRENCE	requirements_property	210	340	requirements_property <= property_definition {property_definition property_definition.name = 'electrical requirement'}}
ELECTROMAGNETIC_RE- QUIREMENT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.name = 'electromagnetic requirement'}}
FABRICATION_THICKNESS_- REQUIREMENT_OCCUR- RENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'fabrication thickness requirement'}}
characterized_class	property_definition.id	41		requirements_property <= property_definition {property_definition (property_definition.id = 'over finished conductor') (property_definition.id = 'over finished dielectric')}

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fabrication_thickness_- requirement_occurrence to tolerance_characteristic (as tolerance)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item {representation_item (representation_item.name = 'tolerance') (representation_item.name = 'minimum tolerance') (representation_item.name = 'maximum tolerance') (representation_item.name = 'plus minus tolerance') (representation_item.name = 'symmetrical tolerance') (representation_item.name = 'statistical tolerance')}</pre>
fabrication_thickness_- requirement_occurrence to range_characteristic (as characterized_class_range)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_ representation -> representation.items[i] -> representation_item {representation_item (representation_item.name = 'range characteristic') representation_item => coordinated_representation_item}</pre>
INSPECTION_REQUIRE- MENT_OCCURRENCE	requirements_property	210		<pre> requirements_property <= property_definition {property_definition}</pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(property_definition.description = 'inspection requirement') (property_definition.description = 'layout land width tolerance requirement') (property_definition.description = 'layout line width tolerance requirement')}</pre>
LAYOUT_LAND_WIDTH_TOL- ERANCE_REQUIREMENT_OC- CURRENCE	requirements_property	210		<pre>requirements_property <= property_definition {property_definition property_definition.description = 'layout land width tolerance requirement'}</pre>
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_negative_deviation)	PATH			<pre>requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum negative deviation'}</pre>
layout_land_width_tolerance_- requirement_occurrence to pitch_class (as mating_feature_pitch_class)	PATH			<pre>requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation {representation.name = 'pitch class'}</pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
LAYOUT_LINE_WIDTH_TOLERANCE_REQUIREMENT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'layout line width tolerance requirement'}
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_negative_deviation)	PATH			requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation representation.items[i] -> {representation_item representation_item.name = 'maximum negative deviation'}
layout_land_width_tolerance_- requirement_occurrence to length_data_element (as maximum_positive_deviation)	PATH			representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
layout_line_width_tolerance_- requirement_occurrence to line_width_class (as tolerance_line_width_class)	PATH			requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
				requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation_item => styled_item

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
layout_line_width_tolerance_- requirement_occurrence to stratum_technology (as characterized_stratum_- technology)	PATH			<pre> requirements_property <= property_definition <- property_definition_relationship.related_- property_definition property_definition_relationship {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object characterized_object => stratum_technology </pre>
MATERIAL_ELECTRICAL_- CONDUCTIVITY_REQUIRE- MENT_OCCURRENCE	requirements_property	210	340	<pre> requirements_property <= property_definition {property_definition property_definition.name = 'material electrical conductivity classification'} </pre>
electrical_conductivity_character- istic	material_electrical_conductivity_- category	210		<pre> requirements_property classification_assigned_item = requirements_- property classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> {group (group.name = 'conductive') (group.name = 'non conductive') (group.name = 'resistive') (group.name = 'semi conductive')} </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				group => material_electrical_conductivity_category
MECHANICAL_REQUIREMENT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.name = 'mechanical requirement'}}
OPERATIONAL_REQUIREMENT_OCCURRENCE_RELATIONSHIP	property_definition_relationship	45	217	
relationship_type	property_definition_- relationship.name	45		property_definition_relationship {property_definition_relationship.name (property_definition_relationship.name = 'and operation') (property_definition_relationship.name = 'derived from operation') (property_definition_relationship.name = 'or operation')}}}
operational_requirement_- occurrence_relationship to ee_requirement_occurrence (as related_requirement_occurrence)	PATH		340	property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition => requirements_property
operational_requirement_- occurrence_relationship to ee_requirement_occurrence (as relating_requirement_occurrence)	PATH			property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition => requirements_property
PURPOSE_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'purpose'}}
REQUIREMENT_COMPOSITION	property_definition_relationship	45	283	{property_definition_relationship property_definition_relationship.name = 'requirements property composition'}

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
requirement_composition to ee_requirement_occurrence (as element)	PATH		340	property_definition_relationship property_definition_relationship.related_ property_definition -> property_definition => requirements_property
requirement_composition to requirement_group_occurrence (as composition)	PATH			property_definition_relationship property_definition_relationship.relatin_ property_definition -> property_definition => requirements_property => requirements_property_group
REQUIREMENT_DEFINITION	requirement_definition	210		requirement_definition <= product_definition {[product_definition product_definition.format -> product_definition_format product_definition_format.of_product -> product <= product_related_product_category.products[i] product_related_product_category <= product_category product_category.name = 'requirements model'] [product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'requirement') (application_context_element.name = 'design requirement')]]}
REQUIREMENT_GROUP_OC- CURRENCE	requirements_property_group	210	340	requirements_property_group <= requirements_property <= property_definition

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
RULE_BASED_CONSTRAINT_- OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.id = 'rule based constraint'}
constraint_occurrence to ee_rule_version (as design_constraint)	PATH		340	requirements_property <= property_definition <= property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'design_constraint'} property_definition.relationship.relatin_ property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition => rule_definition} product_definition.formation -> product_definition.formation {product_definition.formation.description = 'rule version'}
SHAPE AND EE_PRODUCT_- DEFINITION_BASED_CON- STRAINT_OCCURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'shape and product definition based constraint'}

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shape_and_ee_product_definition_based_constraint_- occurrence to shape_definition (as constraining_shape)	PATH			<pre> requirements_property <= property_definition <- property_definition_representation.definition property_definition_representation.used_- representation -> representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'constraint shape reference'} representation_relationship.rep_2 -> representation {representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition {(characterized_product_definition = product_- definition) (characterized_product_definition = product_- definition => product_definition_with_associated_documents) (characterized_product_definition = product_- definition => physical_unit) (characterized_product_definition = product_- definition => </pre>

Table 90 – Mapping table for requirement_decomposition UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				externally_defined_physical_unit (characterized_product_definition = product_ definition => externally_defined_physical_unit => library_defined_physical_unit) (characterized_product_definition = product_ definition => analytical_model_definition) (characterized_product_definition = product_ definition => footprint_definition) (characterized_product_definition = product_ definition => padstack_definition) (characterized_product_definition = product_ definition => physical_unit_network_definition) (characterized_product_definition = product_ definition => requirement_definition) (characterized_product_definition = product_ definition => stratum) (characterized_product_definition = product_ definition => functional_unit) (characterized_product_definition = product_ definition => functional_unit => externally_defined_functional_unit) (characterized_product_definition = product_ definition => functional_unit => externally_defined_functional_unit =>

Table 90 – Mapping table for requirement_decomposition UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				library_defined_functional_unit}} representation => shape_representation
TEST_REQUIREMENT_OC- CURRENCE	requirements_property	210		requirements_property <= property_definition {property_definition property_definition.description = 'test requirement'}
THERMAL_REQUIREMENT_- OCCURRENCE	requirements_property	210	340	requirements_property <= property_definition {property_definition property_definition.name = 'thermal requirement'}

Table 91 – Mapping table for sequential_laminate_assembly_design UoF

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_MATERIAL_COM- POSITION_RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44		(next_assembly_usage_occurrence {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'})} ([quantified_assembly_component_usage] {next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'assembly_material_composition'}})]
as_required	PATH	41		quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit measure_with_unit.value_component -> measure_value measure_value = descriptive_measure {(descriptive_measure = 'as required') (descriptive_measure = 'quantity specified')}}
assembly_material_composition_- relationship to data_element (as quantity)	PATH			quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
assembly_material_composition_- relationship to material_definition (as component_material)	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'material definition'} </pre>
assembly_material_composition_- relationship to physical_unit_design_view (as assembly)	PATH		338	<pre> next_assembly_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relatin_- product_definition -> product_definition product_definition => physical_unit </pre>
assembly_material_composition_- relationship to security_classification (as security_code)	PATH			<pre> next_assembly_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_- security_classification -> security_classification </pre>
INTERCONNECT_MODULE	physical_unit	210	338, 268	<pre> physical_unit <= product_definition </pre>

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']}</pre>
interconnect_module to assembly_module (as assembly_design_requirement)	PATH			<pre>physical_unit <= product_definition <- product_definition.relationship.related_ - product_definition {product_definition.relationship product_definition.relationship.name = 'assembly_design_requirement'} product_definition.relationship.relatin_ - product_definition -> product_definition {product_definition [product_definition => physical_unit] [product_definition.name = 'assembly module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design']}</pre>
INTERCONNECT_MODULE_- COMPONENT	component_definition	210	338, 268	<pre>component_definition <= product_definition {product_definition [product_definition product_definition.description != 'laminate component'] [product_definition.name = 'interconnect module']}</pre>

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical occurrence'.]
interconnect_module_component to interconnect_module_usage_view (as usage_definition)	PATH		338, 268	<pre> component_definition <= product_definition <- product_definition.relationship.related_ product_definition {product_definition.relationship product_definition.relationship.name = 'instantiated part'} product_definition.relationship product_definition.relationship.relatin_ product_definition -> {product_definition [product_definition.name = 'interconnect module']} [product_definition.frame_of_reference -> product_definition.context <= application_context_element application_context_element.name = 'physical design usage']} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) </pre>
INTERCONNECT_MODULE_- COMPONENT_STRATUM_- BASED_TERMINAL	component_terminal	210	333	<pre> component_terminal <= shape_aspect {shape_aspect shape_aspect.description = 'interconnect module component stratum based terminal'} </pre>

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
interconnect_module_- component_stratum_based_- terminal to interconnect_module_stratum_- based_terminal (as definition)	PATH		333	<pre> component_terminal <= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.description = 'interconnect module component stratum based terminal definition'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect => interconnect_module_stratum_based_terminal interconnect_module_component_surface_feature </pre>
INTERCONNECT_MODULE_- COMPONENT_SURFACE_FEA- TURE	interconnect_module_- component_surface_feature	210	333	<pre> interconnect_module_component_surface_feature <= shape_aspect {shape_aspect.description = 'interconnect module component surface feature'} interconnect_module_component_surface_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'instantiated feature'} shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {(shape_aspect.description = 'interconnect module cavity surface') (shape_aspect.description = 'interconnect module cutout surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module edge segment surface')} </pre>
interconnect_module_- component_surface_feature to interconnect_module_surface_- feature (as definition)	PATH			

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature')}
INTERCONNECT_MODULE_- PRODUCT_ASSEMBLY_- VIEW_FABRICATION_JOINT	sequential_laminate_passage_- based_fabrication_joint	210		sequential_laminate_passage_based_- fabrication_joint <= shape_aspect {shape_aspect.description = 'interconnect module product assembly view fabrication joint'}
interconnect_module_product_- assembly_view_fabrication_joint to interconnect_module_- component_stratum_based_- terminal (as joined_terminal)	PATH		333	sequential_laminate_passage_based_- fabrication_joint <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'interconnect module component stratum based terminal'} shape_aspect => component_terminal

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
INTERCONNECT_MODULE_- STRATUM_BASED_TERMINAL	interconnect_module_stratum_- based_terminal	210	333, 340, 268	interconnect_module_stratum_based_terminal <= interconnect_module_terminal <= shape_aspect {shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition product_definition product_definition.name = 'interconnect module'}
NEXT_HIGHER_ASSEMBLY_- RELATIONSHIP	(next_assembly_usage_- occurrence) ([next_assembly_usage_- occurrence] [quantified_assembly_- component_usage])	44 44 44	213, 214	(next_assembly_usage_occurrence) ([next_assembly_usage_occurrence <=] [quantified_assembly_component_usage <=] assembly_component_usage)
reference_designation	PATH			next_assembly_usage_occurrence <= assembly_component_usage assembly_component_usage.reference_designator
next_higher_assembly_- relationship to data_element (as quantity)	PATH			next_assembly_usage_occurrence <= assembly_component_usage => quantified_assembly_component_usage quantified_assembly_component_usage.quantity -> measure_with_unit

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect }) (component_interface_terminal <= shape_aspect)
PROMISSORY_USAGE_IN_PRODUCT_MODEL_RELATIONSHIP	applied_promissory_usage_in_product_model_assignment	210		{applied_promissory_usage_in_product_model_assignment <= group_assignment group_assignment.assigned_group -> group => promissory_usage_in_product_model} applied_promissory_usage_in_product_model_assignment
promissory_usage_in_product_model_relationship to Ee_product_definition (as component)	PATH			applied_promissory_usage_in_product_model_assignment.items[i] -> promissory_usage_in_product_model_assigned_item promissory_usage_in_product_model_assigned_item = product_definition (product_definition => product_definition.with_associated_documents) (product_definition) (product_definition => evaluation_product_definition) (product_definition => physical_unit) (product_definition => externally_defined_physical_unit) (product_definition => library_defined_physical_unit) (product_definition => analytical_model_definition) (product_definition => footprint_definition)

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (product_definition => padstack_definition) (product_definition => physical_unit_network_definition) (product_definition => requirement_definition) (product_definition => functional_unit) (product_definition => externally_defined_functional_unit) (product_definition => library_defined_functional_unit) applied_promissory_usage_in_product_model_- assignment applied_promissory_usage_in_product_model_- assignment.items[i] -> promissory_usage_in_product_model_assigned_- item promissory_usage_in_product_model_assigned_- item = product_concept applied_promissory_usage_in_product_model_- assignment security_classification_assigned_item = applied_promissory_usage_in_product_model_- assignment security_classification_assigned_item <- applied_security_classification_- assignment.items[i] applied_security_classification_assignment <= security_classification_assignment </pre>
promissory_usage_in_product_- model_relationship to ee_product_model (as product_model)	PATH			
promissory_usage_in_product_- model_relationship to security_classification (as security_code)	PATH			

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				security_classification_assignment.assigned_- security_classification -> security_classification
PROMISSORY_USAGE_RELATIONSHIP	promissory_usage_occurrence	44	275, 276	
promissory_usage_relationship to physical_unit (as component)	PATH		354, 339, 338	promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element (application_context_element.name = 'physical design') (application_context_element.name = 'physical design usage')} product_definition => (physical_unit) (physical_unit => externally_defined_physical_unit) (physical_unit => externally_defined_physical_unit => library_defined_physical_unit) library_defined_physical_unit
promissory_usage_relationship to physical_unit_design_view (as assembly)	PATH			promissory_usage_occurrence <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition ->

Table 91 – Mapping table for sequential_laminate_assembly_design UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design'} product_definition => physical_unit
promissory_usage_relationship to security_classification (as security_code)	PATH			promissory_usage_occurrence <= assembly_component_usage security_classification_assigned_item = assembly_component_usage security_classification_assigned_item <= applied_security_classification_ assignment.items[i] applied_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_ security_classification -> security_classification

Table 92 – Mapping table for shape UoF

Application element	AIM element	Source	Rules	Reference path
AXIS_PLACEMENT	(axis2_placement_2d) (axis2_placement_3d)	42 42		(axis2_placement_2d) (axis2_placement_3d)
axis_placement to cartesian_point (as translation)	PATH			(axis2_placement_2d <= placement placement.location ->) (axis2_placement_3d <= placement placement.location ->) cartesian_point
axis_placement to direction (as reference_direction)	PATH			(axis2_placement_2d axis2_placement_2d.ref_direction ->) (axis2_placement_3d axis2_placement_3d.ref_direction ->) direction
AXIS_PLACEMENT_2D	axis2_placement_2d	42		
AXIS_PLACEMENT_3D	axis2_placement_3d	42		
BOOLEAN_CSG_RESULT	boolean_result	42		
csg_operator	boolean_operator	42		boolean_result boolean_result.operator -> boolean_operator {boolean_operator (boolean_operator = .UNION.) (boolean_operator = .INTERSECTION.) (boolean_operator = .DIFFERENCE.)}}
boolean_csg_result to boolean_csg_result (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = boolean_result boolean_result

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
boolean_csg_result to boolean_csg_result (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = boolean_result boolean_result
boolean_csg_result to bound_volume_shape (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand (boolean_operand = half_space_solid) (boolean_operand = solid_model)
boolean_csg_result to bound_volume_shape (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand (boolean_operand = half_space_solid) (boolean_operand = solid_model)
boolean_csg_result to csg_primitive (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = csg_primitive
boolean_csg_result to csg_primitive (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = csg_primitive
boolean_csg_result to primitive_2d (as operand_1)	PATH			boolean_result boolean_result.first_operand -> boolean_operand boolean_operand = csg_primitive csg_primitive = primitive_2d
boolean_csg_result to primitive_2d (as operand_2)	PATH			boolean_result boolean_result.second_operand -> boolean_operand boolean_operand = csg_primitive csg_primitive = primitive_2d

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
BOUND_VOLUME_SHAPE	shape_representation	41	111	{shape_representation <= representation representation.name = '3d bound volume shape'}
BOUNDING_CURVE	trimmed_curve	42		
bounding_curve to external_definition (as mapping_function)	PATH		109	<pre> trimmed_curve <= bounded_curve <= curve <= geometric_representation_item <= representation_item <- representation.items[i] representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object {characterized_object characterized_object.name = 'mapping function'} characterized_object => external_definition trimmed_curve trimmed_curve.basis_curve -> curve => trimmed_curve </pre>
bounding_curve to trimmed_line (as basis_curve)	PATH			
CARTESIAN_COORDINATE_- SYSTEM	([geometric_representation_- context] [global_unit_assigned_context] [global_uncertainty_assigned_- context])	42 41 43	24	

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
angle_unit	(plane_angle_unit) (solid_angle_unit)	41 41		<pre> global_unit_assigned_context global_unit_assigned_context.units[i] -> unit unit = named_unit {named_unit => (si_unit si_unit.name -> si_unit.name si_unit_name = .RADIAN.) (conversion_based_unit conversion_based_unit.name = 'degree')} named_unit => (plane_angle_unit) (solid_angle_unit) </pre>
length_dynamic_range	PATH			<pre> global_uncertainty_assigned_context global_uncertainty_assigned_ - context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit {measure_with_unit => length_measure_with_unit} {measure_with_unit.unit_component -> unit = named_unit named_unit => ratio_unit} measure_with_unit.value_component -> measure_value = positive_ratio_measure </pre>
dimensionality	geometric_representation_- context.coordinate_space_- dimension	42		<pre> {geometric_representation_context geometric_representation_context.coordinate_- space_dimension (geometric_representation_context.coordinate_- space_dimension = 2) (geometric_representation_context.coordinate_- space_dimension = 3)} </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
length_unit	(si_unit.name) (conversion_based_unit.name)	41 41		global_unit_assigned_context global_unit_assigned_context.units[i] -> unit unit = named_unit named_unit {named_unit => length_unit} (named_unit => si_unit si_unit.name) (named_unit => conversion_based_unit conversion_based_unit.name) global_unit_assigned_context global_unit_assigned_context.units[i] -> unit unit = named_unit named_unit => si_unit si_unit.prefix {(si_unit.prefix = .exa.) (si_unit.prefix = .peta.) (si_unit.prefix = .tera.) (si_unit.prefix = .giga.) (si_unit.prefix = .mega.) (si_unit.prefix = .kilo.) (si_unit.prefix = .hecto.) (si_unit.prefix = .deca.) (si_unit.prefix = .deci.) (si_unit.prefix = .centi.) (si_unit.prefix = .milli.) (si_unit.prefix = .micro.) (si_unit.prefix = .nano.) (si_unit.prefix = .pico.)
length_unit_prefix	si_unit.prefix	41		

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(si_unit.prefix = .femto.) (si_unit.prefix = .atto.)}
cartesian_coordinate_system to angle_data_element (as angular_uncertainty)	PATH			global_uncertainty_assigned_context global_uncertainty_assigned_ context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => plane_angle_measure_with_unit
cartesian_coordinate_system to length_data_element (as length_uncertainty)	PATH			global_uncertainty_assigned_context global_uncertainty_assigned_ context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit
CARTESIAN_POINT	cartesian_point	42		
CARTESIAN_TRANSFORMATION_OPERATOR_2D	cartesian_transformation_ operator_2d	42	25	
scale	cartesian_transformation_ operator.scale	42		
cartesian_transformation_ operator_2d to cartesian_point (as translation)	PATH			cartesian_transformation_operator_2d <= cartesian_transformation_operator cartesian_transformation_operator.local_origin -> cartesian_point
cartesian_transformation_ operator_2d to direction (as axis_1)	PATH			cartesian_transformation_operator_2d <= cartesian_transformation_operator cartesian_transformation_operator.axis1 -> direction
cartesian_transformation_ operator_2d to direction (as axis_2)	PATH			cartesian_transformation_operator_2d <= cartesian_transformation_operator cartesian_transformation_operator.axis2 -> direction

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
CIRCULAR_AREA	circular_area	42		
circular_area to cartesian_point (as centre)	PATH			circular_area circular_area.centre -> cartesian_point
circular_area to length_data_element (as radius)	PATH			circular_area circular_area.radius -> positive_length_measure
CLOSED_CURVE	composite_curve	42		{composite_curve composite_curve.closed_curve = .TRUE.}
COMPOSITE_CURVE	composite_curve	42		
composite_curve to composite_curve_segment (as curve_segment)	PATH			composite_curve.segments[i] -> composite_curve_segment
COMPOSITE_CURVE_SEGMENT	composite_curve_segment	42		
composite_curve_segment to curve (as segment_curve)	PATH			composite_curve_segment.parent_curve -> curve
CONNECTION_ZONE	shape_aspect	41	347, 66	{shape_aspect shape_aspect.description = 'connection zone'}
connection_zone to ee_colour (as representation_colour)	PATH			shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition property_definition <- property_definition.definition property_definition.representation.definition property_definition.representation.used_ representation ->

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>representation { representation => shape_representation representation.name = 'zone shape' property_definition <- property_definition_representation.definition property_definition_representation => {property_definition_representation} shape_definition_representation property_definition_representation.used_ representation -> representation {[representation => shape_representation] [representation representation.context_of_items -> representation_context representation_context.context_type = 'connection zone colour']} representation.items[i] -> representation_item representation_item => styled_item {styled_item styled_item.item -> representation_item => geometric_representation_item} styled_item styled_item.styles[i] -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = fill_area_style</pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style.colour fill_area_style.colour.fill_colour -> colour => colour_specification => colour_rgb
CSG_2D_SHAPE	csg_2d_shape_representation	210		
CURVE	curve	42		
DIRECTION	direction	42		
ELLIPTIC_AREA	elliptic_area	42		
elliptic_area to axis_placement_2d (as position)	PATH			elliptic_area elliptic_area.position -> axis2_placement_2d
elliptic_area to length_data_element (as semi_axis_1)	PATH			elliptic_area elliptic_area.semi_axis_1 -> positive_length_measure
elliptic_area to length_data_element (as semi_axis_2)	PATH			elliptic_area elliptic_area.semi_axis_2 -> positive_length_measure
EXTERNALLY_DEFINED_- HATCH_STYLE	externally_defined_hatch_style	46		
EXTERNALLY_DEFINED_TILE	externally_defined_tile	46		
EXTERNALLY_DEFINED_- TILE_STYLE	externally_defined_tile_style	46		
FEATURE_SHAPE_DEFINITION	shape_representation	43		{!shape_representation <= representation <- property_definition_representation.used_ representation

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.product_definitional = .FALSE.}}
FILL_AREA		46		
fill_area to closed_curve (as boundaries)	annotation_fill_area PATH			annotation_fill_area annotation_fill_area.boundaries[i] -> curve => bounded_curve => composite_curve {composite_curve.closed_curve = .TRUE.}
FILL_AREA_OCCURRENCE		46	170	
fill_area_occurrence to cartesian_point (as fill_style_target)	PATH			annotation_fill_area_occurrence annotation_fill_area_occurrence.fill_style_- target -> point => cartesian_point
fill_area_occurrence to curve_style (as boundary_style)	PATH			(annotation_fill_area_occurrence <= annotation_occurrence <= styled_item styled_item.item -> representation_item =>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> geometric_representation_item => annotation_fill_area annotation_fill_area.boundaries[i] -> curve <= geometric_representation_item <= representation_item <- styled_item.item styled_item styled_item.styles[i] -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = curve_style curve_style (annotation_fill_area_occurrence <= annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => annotation_fill_area annotation_fill_area.boundaries[i] -> curve <= geometric_representation_item <= representation_item <- representation.items[i] representation <- representation_relationship.rep_1 {representation_relationship representation_relationship.name = 'boundary style'} representation_relationship.rep_2 -> representation </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation representation.name = 'curve style parameters'}
fill_area_occurrence to fill_area (as filled_area)	PATH			annotation_fill_area_occurrence <= annotation_occurrence <= styled_item -> representation_item => geometric_representation_item => annotation_fill_area
fill_area_occurrence to fill_area_style (as style)	PATH			annotation_fill_area_occurrence <= annotation_occurrence <= styled_item -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = fill_area_style fill_area_style
FILL_AREA_STYLE	fill_area_style	46	171, 172	
name	fill_area_style.name	46		
fill_area_style to externally_defined_hatch_style (as fill_style)	PATH			fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = externally_defined_hatch_ style externally_defined_hatch_style
fill_area_style to externally_defined_tile_style (as fill_style)	PATH			fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = externally_defined_tile_ style externally_defined_tile_style

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fill_area_style to fill_area_style_hatching (as fill_style)	PATH			fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style_hatching fill_area_style_hatching
fill_area_style to fill_area_style_tile (as fill_style)	PATH			fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style_tiles fill_area_style_tiles
fill_area_style to solid_fill_style (as fill_style)	PATH			fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style_colour fill_area_style_colour
FILL_AREA_STYLE_HATCH- ING	fill_area_style_hatching	46		
fill_area_style_hatching to angle_data_element (as hatch_line_angle)	fill_area_style_hatching.hatch_ line_angle	46		
fill_area_style_hatching to cartesian_point (as pattern_start)	PATH			fill_area_style_hatching fill_area_style_hatching.pattern_start -> cartesian_point
fill_area_style_hatching to cartesian_point (as point_of_reference_hatch_line)	PATH			fill_area_style_hatching fill_area_style_hatching.point_of_reference_ hatch_line -> cartesian_point
fill_area_style_hatching to curve_style (as hatch_line_appearance)	PATH			fill_area_style_hatching fill_area_style_hatching.hatch_line_appearance -> curve_style
fill_area_style_hatching to vector (as start_of_next_hatch_line)	PATH			fill_area_style_hatching fill_area_style_hatching.start_of_next_hatch_ line -> one_direction_repeat_factor one_direction_repeat_factor.repeat_factor -> vector

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
FILL_AREA_STYLE_TILE	fill_area_style_tiles	46		
tiling_scale	fill_area_style_tiles.tiling_scale	46		
fill_area_style_tile to externally_defined_tile (as tile)	PATH			fill_area_style_tiles fill_area_style_tiles.tiles[i] -> fill_area_style_tile_shape_select fill_area_style_tile_shape_select = externally_defined_tile externally_defined_tile
fill_area_style_tile to fill_area_- style_tile_curve_with_style (as tile)	PATH			fill_area_style_tiles fill_area_style_tiles.tiles[i] -> fill_area_style_tile_shape_select fill_area_style_tile_shape_select = fill_- area_style_tile_curve_with_style fill_area_style_tile_curve_with_style
fill_area_style_tile to fill_area_style_tile_solid_region (as tile)	PATH			fill_area_style_tiles fill_area_style_tiles.tiles[i] -> fill_area_style_tile_shape_select fill_area_style_tile_shape_select = fill_- area_style_tile_coloured_region fill_area_style_tile_coloured_region
fill_area_style_tile to fill_area_- style_tile_symbol_with_style (as tile)	PATH			fill_area_style_tiles fill_area_style_tiles.tiles[i] -> fill_area_style_tile_shape_select fill_area_style_tile_shape_select = fill_- area_style_tile_symbol_with_style fill_area_style_tile_symbol_with_style
fill_area_style_tile to vector (as first_direction_repeat_factor)	PATH			fill_area_style_tiles fill_area_style_tiles.tiling_pattern -> two_direction_repeat_factor two_direction_repeat_factor <= one_direction_repeat_factor one_direction_repeat_factor.repeat_factor -> vector

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
fill_area_style_tile to vector (as second_direction_repeat_factor)	PATH			fill_area_style_tiles fill_area_style_tiles.tiling_pattern -> two_direction_repeat_factor two_direction_repeat_factor.second_repeat_ factor -> vector
FILL_AREA_STYLE_TILE_- CURVE_WITH_STYLE	fill_area_style_tile_curve_with_ style	46		
FILL_AREA_STYLE_TILE_- SOLID_REGION	fill_area_style_tile_coloured_ region	46		
FILL_AREA_STYLE_TILE_- SYMBOL_WITH_STYLE	fill_area_style_tile_symbol_with_ style	46		
GEOMETRIC REPRESENTA- TION_RELATIONSHIP	(shape_representation_ relationship) (mapped_item) (representation_relationship_ with_transformation)	41 43 43		{shape_representation_relationship <= representation_relationship {(representation_relationship.name = 'component part planar shape') (representation_relationship.name = 'component part non planar shape') (representation_relationship.name = 'generic relationship') (representation_relationship.name = 'component part 3d shape')} {representation_relationship (representation_relationship.description = 'component to part') (representation_relationship.description = 'non feature shape annotation')}}} (mapped_item {mapped_item <= representation_item.name = 'non feature shape annotation'}) (representation_relationship_with_ transformation <=

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_relationship representation_relationship.description = 'non feature shape annotation'}
geometric_representation_- relationship to shape_definition (as shape_1)	PATH			(shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation {representation => shape_representation} (representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation) (mapped_item mapped_item.mapping_source -> representation_map {representation_map.mapping_origin -> representation_item {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement (placement => axis2_placement_2d) (placement => axis2_placement_3d)} representation_map.mapped_representation -> representation)
geometric_representation_- relationship to shape_definition (as shape_2)	PATH			(shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> representation {representation => representation

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> shape_representation{ (representation_relationship_with_ transformation <= representation_relationship representation_relationship.rep_2 -> representation => shape_representation) (mapped_item <= representation_item <- representation.items[i] representation => shape_representation) </pre>
MANIFOLD_SUBSURFACE_- SHAPE	manifold_subsurface_shape_- representation	521		
NON_FEATURE_SHAPE_DEFINITION	shape_representation	43		<pre> {shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}} </pre>
non_feature_shape_definition to feature_shape_definition (as model_shape)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'} </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
non_feature_shape_definition to non_feature_shape_element (as associated_element)	PATH			representation_relationship.rep_1 -> representation => shape_representation shape_representation <= representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect shape_aspect.product_definitional = .FALSE.}
PLANAR_PROJECTED_SHAPE	shape_representation	41	111, 247	{shape_representation <= representation representation.name = 'planar projected shape'}
PLANAR_SHAPE	shape_representation	41		{shape_representation <= representation.context_of_items -> representation_context => geometric_representation_context.coordinate_ space_dimension = 2}
POLYGONAL_AREA	polygonal_area	42		
polygonal_area to cartesian_point (as bounds)	PATH			polygonal_area polygonal_area.bounds[i] -> cartesian_point

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
PRE_DEFINED_DATUM_- AXIS_3D_SYMBOL	shape_representation	41		shape_representation <= {representation representation.name = '3d bound volume shape'} {representation representation.description = 'pre defined datum axis 3d symbol'} representation
PRE_DEFINED_DATUM_- PLANE_3D_SYMBOL	shape_representation	41		shape_representation <= {representation representation.name = '3d bound volume shape'} {representation representation.description = 'pre defined datum plane 3d symbol'} representation
PRE_DEFINED_PARALLEL_- DATUM_AXIS_2D_SYMBOL	shape_representation	41		shape_representation <= {representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined parallel datum axis 2d symbol'} representation
PRE_DEFINED_PARALLEL_- DATUM_AXIS_SYMBOL_3D_- 2D_RELATIONSHIP	pre_defined_parallel_datum_- axis_symbol_3d_2d_relationship	210		{pre_defined_parallel_datum_axis_symbol_3d_- 2d_relationship <= volume_shape_intersection} pre_defined_parallel_datum_axis_symbol_3d_2d_- relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined parallel datum axis 2d symbol'} representation => shape_representation
pre_defined_parallel_datum_- axis_symbol_3d_2d_relationship to pre_defined_parallel_datum_- axis_2d_symbol (as surface_intersection_result)	PATH			

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
pre_defined_parallel_datum_- axis_symbol_3d_2d_relationship to pre_defined_datum_axis_3d_- symbol (as projected_volume)	PATH			pre_defined_parallel_datum_axis_symbol_3d_2d_- relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation representation.description = 'pre defined datum axis 3d symbol'}
PRE_DEFINED_PERPENDICU- LAR_DATUM_AXIS_2D_SYM- BOL	shape_representation	41		{representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined perpendicular datum axis 2d symbol'} representation
PRE_DEFINED_PERPENDIC- ULAR_DATUM_AXIS_SYM- BOL_3D_2D_RELATIONSHIP	pre_defined_perpendicular_- datum_axis_symbol_3d_2d_- relationship	210		{pre_defined_perpendicular_datum_axis_symbol_- 3d_2d_relationship <= volume_shape_intersection}
pre_defined_perpendicular_- datum_axis_symbol_3d_2d_- relationship to pre_defined_perpendicular_- datum_axis_2d_symbol (as surface_intersection_result)	PATH			pre_defined_perpendicular_datum_axis_symbol_- 3d_2d_relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined perpendicular datum axis 2d symbol'} representation => shape_representation

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
pre_defined_perpendicular_- datum_axis_symbol_3d_2d_- relationship to pre_defined_datum_axis_3d_- symbol (as projected_volume)	PATH			pre_defined_perpendicular_- 3d_2d_relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation representation.description = 'pre defined datum axis 3d symbol'}
PRE_DEFINED_PERPENDIC- ULAR_DATUM_PLANE_2D_- SYMBOL	shape_representation	41		{representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined perpendicular datum plane 2d symbol'}
PRE_DEFINED_PERPENDIC- ULAR_DATUM_PLANE_SYM- BOL_3D_2D_RELATIONSHIP	pre_defined_perpendicular_- datum_plane_symbol_3d_2d_- relationship	210		{pre_defined_perpendicular_- symbol_3d_2d_relationship <= volume_shape_intersection}
pre_defined_perpendicular_- datum_plane_symbol_3d_2d_- relationship to pre_defined_perpendicular_- datum_plane_2d_symbol (as surface_intersection_result)	PATH			pre_defined_perpendicular_- 3d_2d_relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} {representation representation.description = 'pre defined perpendicular datum plane 2d symbol'}
				shape_representation

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
pre_defined_perpendicular_- datum_plane_symbol_3d_2d_- relationship to pre_defined_datum_plane_3d_- symbol (as projected_volume)	PATH			pre_defined_perpendicular_datum_plane_symbol_- 3d_2d_relationship <= volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> {representation representation.description = 'pre defined datum plane 3d symbol'}
PRIMITIVE_2D	primitive_2d	42		{representation representation.name = '3d bound volume shape'} representation => shape_representation
RECTANGULAR_AREA	rectangular_area	42		
rectangular_area to axis_placement_2d (as position)	PATH			rectangular_area rectangular_area.position -> axis2_placement_2d
rectangular_area to length_data_element (as x)	PATH			rectangular_area rectangular_area.x-> positive_length_measure
rectangular_area to length_data_element (as y)	PATH			rectangular_area rectangular_area.y-> positive_length_measure
SHAPE_COMPOSITION_RELATIONSHIP	(representation_relationship_- with_transformation) (mapped_item)	43 43		(mapped_item {mapped_item <= representation_item.name = 'non feature shape annotation'}) (representation_relationship_with_- transformation <= {representation_relationship representation_relationship.description = 'non feature shape annotation'})

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
shape_composition_relationship to axis_placement (as related_feature_shape_placement)	PATH			(mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement)
shape_composition_relationship to cartesian_transformation_operator_2d (as related_feature_shape_placement)	PATH			(representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = functionally_defined_- transformation functionally_defined_transformation => cartesian_transformation_operator)
shape_composition_relationship to shape_definition (as resultant_feature_shape)	PATH			(representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> representation => shape_representation) (mapped_item <= representation_item <= representation.items[i] representation => shape_representation)
shape_composition_relationship to shape_definition (as related_feature_shape)	PATH			(representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation)

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (mapped_item mapped_item.mapping_source -> representation_map {representation_map.mapping_origin -> representation_item } representation_item {representation_item.name = 'origin'} representation_item => geometric_representation_item => placement (placement => axis2_placement_2d) (placement => axis2_placement_3d)} representation_map.mapped_representation -> representation) </pre>
SHAPE_DEFINITION	shape_representation	43		
shape_definition to cartesian_coordinate_system (as geometric_context)	PATH			<pre> shape_representation <= representation representation.context_of_items -> representation_context => ([geometric_representation_context] [global_unit_assigned_context]) ([geometric_representation_context] [global_unit_assigned_context] [global_uncertainty_assigned_context]) </pre>
shape_definition to shape_item (as element)	PATH			<pre> shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item {(geometric_representation_item => </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>boolean_result) (geometric_representation_item => placement placement.location -> cartesian_point) (geometric_representation_item => curve) (geometric_representation_item => primitive_2d)}</pre>
SHAPE_ITEM	geometric_representation_item	42		
SOLID_FILL_STYLE	fill_area_style_colour	46		
TRIMMED_LINE	trimmed_curve	42		
trimmed_line to cartesian_point (as start_point)	PATH			<pre>trimmed_curve trimmed_curve.trim_1 -> cartesian_point</pre>
trimmed_line to cartesian_point (as end_point)	PATH			<pre>trimmed_curve trimmed_curve.trim_1 -> cartesian_point</pre>
VECTOR	vector	42		
VIEW_LEVEL_NON_FEATURE_SHAPE_DEFINITION	shape_representation	43		
view_level_non_feature_shape_definition to part_template_3d_shape (as model_shape)	PATH			<pre>shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'}</pre> <pre>representation_relationship.rep_1 -></pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]} representation => shape_representation shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'} representation_relationship.rep_1 -> {[representation => shape_representation] [representation.name = '3d bound volume shape']} [representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_- space_dimension = 3] </pre>
view_level_non_feature_shape - definition to physical_unit_3d_shape (as model_shape)	PATH			

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>[representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit]]</pre>
view_level_non_feature_shape_- definition to part_template_non_planar_2d_- shape (as model_shape)	PATH			<pre>shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'} representation_relationship.rep_1 -> {representation representation.name = 'part template non planar 2d shape'} [representation => shape_representation => manifold_surface_shape_representation]]</pre>
view_level_non_feature_shape_- definition to part_template_planar_shape (as model_shape)	PATH			<pre>shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'} representation_relationship.rep_1 -></pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {[representation => shape_representation] [representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect => part_template_definition]] </pre>
view_level_non_feature_shape_- definition to physical_unit_planar_shape (as model_shape)	PATH			<pre> shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship representation_relationship.name = 'model shape'} representation_relationship.rep_1 -> {[representation => shape_representation] [representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ - representation </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => physical_unit[] </pre>
VOLUME_SHAPE_INTERSECTION	volume_shape_intersection	210		<pre> {volume_shape_intersection <= [representation_relationship_with_ transformation representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> transformation transformation = item_defined_transformation] [shape_representation_relationship] [representation] [shape_representation_relationship <= representation_relationship representation_relationship.name = 'volume shape_intersection']} </pre>
volume_shape_intersection to axis_placement_2d (as planar_shape_origin)	PATH			<pre> volume_shape_intersection <= representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item { representation_item representation_item.name = 'origin' } representation_item => geometric_representation_item => placement => axis2_placement_2d </pre>
<pre> volume_shape_intersection to axis_placement_3d (as orientation_intersection_plane) </pre>	PATH			<pre> volume_shape_intersection <= representation_relationship_with_ transformation representation_relationship_with_ transformation.transformation_operator -> transformation transformation = item_defined.transformation item_defined.transformation.transformation_item_2 -> representation_item { representation_item representation_item.name = 'orientation intersection plane' } representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>
<pre> volume_shape_intersection to bound_volume_shape (as projected_volume) </pre>	PATH			<pre> volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_2 -> representation { representation.name = '3d bound volume shape' } </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				representation => shape_representation
volume_shape_intersection to planar_projected_shape (as surface_intersection_result)	PATH			volume_shape_intersection <= shape_representation_relationship <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} representation => shape_representation
VOLUME_SHAPE_PLANE_- PROJECTION	representation_relationship_with_- transformation	43		{representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'volume shape plane projection'} representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement => axis2_placement_2d
volume_shape_plane_projection to axis_placement_2d (as planar_shape_origin)	PATH			representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_2 -> representation_item
volume_shape_plane_projection to axis_placement_3d (as orientation_viewing_plane)	PATH			representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_2 -> representation_item

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'orientation viewing plane'} representation_item => geometric_representation_item => surface => elementary_surface {elementary_surface => plane} elementary_surface.position -> axis2_placement_3d </pre>
volume_shape_plane_projection to axis_placement_3d (as viewing_plane)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = '3d bound volume shape'} representation representation.items[i] -> representation_item => geometric_representation_item => geometric_set geometric_set.elements[i] -> geometric_set_select = surface {surface <= geometric_representation_item <= representation_item representation_item.name = 'viewing plane'} surface => elementary_surface {elementary_surface => plane} elementary_surface.position -> axis2_placement_3d </pre>

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
volume_shape_plane_projection to bound_volume_shape (as projected_volume)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = '3d bound volume shape'} representation => shape_representation
volume_shape_plane_projection to planar_projected_shape (as surface_projection_result)	PATH			representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} representation => shape_representation
VOLUME_SHAPE_POINT_- PROJECTION	representation_relationship_with_- transformation	43		{representation_relationship_with_- transformation representation_relationship_with_- transformation.name = 'volume shape point projection'}
volume_shape_point_projection to axis_placement_2d (as planar_shape_origin)	PATH			representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_1 -> representation_item

Table 92 – Mapping table for shape UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> {representation_item representation_item.name = 'origin'} representation_item => geometric_representation_item => placement => axis2_placement_2d </pre>
volume_shape_point_projection to axis_placement_3d (as orientation_viewing_plane)	PATH			<pre> representation_relationship_with_- transformation representation_relationship_with_- transformation.transformation_operator -> transformation transformation = item_defined_transformation item_defined_transformation.transformation_item_2 -> representation_item {representation_item representation_item.name = 'orientation viewing plane'} representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>
volume_shape_point_projection to cartesian_point (as viewing_point)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = '3d bound volume shape'} representation representation.items[i] -> (representation_item => geometric_representation_item => geometric_set geometric_set.elements[i] -> </pre>

Table 92 – Mapping table for shape UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre> point => cartesian_point {cartesian_point <= geometric_representation_item <= representation_item representation_item.name = 'viewing point'}} (representation_item {representation_item representation_item.name = 'viewing point'} representation_item => geometric_representation_item => cartesian_point) </pre>
volume_shape_point_projection to bound_volume_shape (as projected_volume)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_2 -> {representation representation.name = '3d bound volume shape'} representation => shape_representation </pre>
volume_shape_point_projection to planar_projected_shape (as surface_projection_result)	PATH			<pre> representation_relationship_with_- transformation <= representation_relationship representation_relationship.rep_1 -> {representation representation.name = 'planar projected shape'} representation => shape_representation </pre>

Table 93 – Mapping table for signal UoF

Application element	AIM element	Source	Rules	Reference path
CHARACTERISTIC_DATA_SET	data_set_representation_item	210		data_set_representation_item <= compound_representation_item {compound_representation_item compound_representation_item.item_element -> compound_item_definition = set_ representation_item} compound_representation_item <= representation_item
CHARACTERISTIC_DATA_TABLE	table_representation_item	210		table_representation_item <= compound_representation_item {compound_representation_item compound_representation_item.item_element -> compound_item_definition = list_ representation_item} compound_representation_item <= representation_item
COMPOSITE_SIGNAL_PROPERTY	property_definition	41		{property_definition property_definition.name = 'composite signal property'}
COMPOSITE_SIGNAL_PROPERTY_RELATIONSHIP	composite_signal_property_- relationship	210		{[composite_signal_property_relationship <= property_definition_relationship] [composite_signal_property_relationship <= property_definition]}
composite_signal_property_- relationship to composite_signal_property (as composition)	PATH		56	composite_signal_property_relationship <= property_definition_relationship property_definition_relationship.relatiing_- property_definition -> property_definition {property_definition property_definition.name = 'composite signal property'}

Table 93 – Mapping table for signal UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
composite_signal_property_- relationship to external_definition (as aggregation_operation)	PATH		56	<pre> composite_signal_property_relationship <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- object {characterized_object characterized_object.description = 'aggregate operation'} characterized_object => external_definition property_definition.relationship property_definition.relationship.related_- property_definition -> property_definition {property_definition property_definition.name = 'signal property'} </pre>
composite_signal_property_- relationship to signal_property (as signal_property_structural_- element)	PATH			
SIGNAL	signal	210		<pre> signal <= characterized_object </pre>
signal to external_definition (as signal_category)	PATH			<pre> signal document_assigned_item = signal document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference document_reference.source = 'signal category'} document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'signal category'} </pre>

Table 93 – Mapping table for signal UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
SIGNAL_PROPERTY	property_definition	41	310	<pre> {{(property_definition property_definition.name = 'signal property') (property_definition property_definition.name = 'composite signal property')}} property_definition (property_definition.definition -> characterized_definition = characterized_ object characterized_object => signal {signal classification_assigned_item = signal classification_assigned_item <- applied_classification_assignment.items[i] applied_classification_assignment <= group_assignment group_assignment.assigned_group -> group {group.description = 'signal property category'} group => signal_category}} (document_assigned_item = property_definition document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'signal property category'}} </pre>
signal_property to external_definition (as signal_property_category)	PATH			

Table 93 – Mapping table for signal UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
signal_property to characteristic_data_table (as signal_property_representation)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item => compound_representation_item => table_representation_item
signal_property to signal (as characterized_signal)	PATH			property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => signal

Table 94 – Mapping table for solid_of_linear_extrusion UoF

Application element	AIM element	Source	Rules	Reference path
SOLID_OF_LINEAR_EXTRUSION	extruded_area_solid	42		{extruded_area_solid <= swept_area_solid swept_area_solid.swept_area -> curve_bounded_surface [curve_bounded_surface.basis_surface -> surface => elementary_surface => plane] [curve_bounded_surface.boundaries[i] -> boundary_curve <= composite_curve_on_surface <= composite_curve composite_curve.segments[i] -> composite_curve_segment composite_curve_segment.parent_curve -> curve => bounded_curve => trimmed_curve trimmed_curve.basis_curve -> curve => (line) (conic)]}

Table 95 – Mapping table for source_control UoF

Application element	AIM element	Source	Rules	Reference path
CERTIFICATION	certification	41	99, 28	
certification_number	certification.name	41		
certification to date (as certification_date)	PATH		27, 102, 289, 350	certification date_assigned_item = certification date_assigned_item <- applied_date_assignment.items[i] {date_assignment date_assignment.role -> date_role date_role.name = 'certification date'} date_assignment date_assignment.assigned_date -> date => calendar_date
certification to date_and_time (as certification_date)	PATH		27, 103	certification date_and_time_assigned_item = certification date_and_time_assigned_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= {date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'certification date'} date_and_time_assignment date_and_time_assignment.assigned_date_and_ time -> date_and_time

Table 95 – Mapping table for source_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
certification to ee_approval (as certification_approval)	PATH		13	certification approval_assigned_item = certification approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
certification to supplied_product_version (as certified_product_version)	PATH			certification <- certification_assignment.assigned_- certification certification_assignment => applied_certification_assignment applied_certification_assignment.items[i] -> certification_assigned_item certification_assigned_item = supplied_part_- relationship supplied_part_relationship <= product_definition_relationship product_definition_relationship.relatiing_- product_definition -> product_definition product_definition.formation -> product_- definition_formation product_definition_formation => product_definition_formation_with_specified_- source
ORGANIZATION	organization	41	219	
address	organizational_address	41	346	organization <- organizational_address.organizations[i] organizational_address
identifier	organization.id	41		
SUPPLIED_PRODUCT_VERSION	product_definition_formation_- with_specified_source	41	357	

Table 95 – Mapping table for source_control UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
supplier_part_number	product.id	41	268, 339	product_definition_formation_with_specified_ source <= {product_definition_formation <- product_definition.formation product_definition <- product_definition_relationship.related_ product_definition product_definition_relationship => supplied_part_relationship} product_definition_formation product_definition_formation.of_product -> product product.id
supplied_product_version to ee_approval (as supplied_product_version - approval)	PATH			product_definition_formation_with_specified_ source approval_assigned_item = product_definition_ formation_with_specified_source approval_assigned_item <- applied_approval_assignment.items[i] applied_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval
supplied_product_version to ee_product_version (as supplied_product_version)	PATH			product_definition_formation_with_specified_ source <= product_definition_formation <- product_definition_formation_ relationship.related_product_definition_ formation product_definition_formation_relationship {product_definition_formation_relationship product_definition_formation_relationship.name = 'supplied product version'}

Table 95 – Mapping table for source_control UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation_- relationship.relation_product_definition_- formation -> product_definition_formation
supplied_product_version to organization (as supplier)	PATH			product_definition_formation_with_specified_- source organization_assigned_item = product_- definition_formation_with_specified_source organization_assigned_item <- applied_organization_assignment.items[i] applied_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'supplier'} organization_assignment organization_assignment.assigned_organization -> organization

Table 96 – Mapping table for stratum UoF

Application element	AIM element	Source	Rules	Reference path
DESIGN_LAYER_STRATUM	stratum	210	265, 266, 338, 263, 264, 268, 267	stratum <= product_definition {product_definition product_definition.name = 'design layer'}
primary_design_layer_stratum	product_definition.description	41	265, 266, 338	stratum <= product_definition {product_definition.description = 'primary design layer stratum' (product_definition.description = 'non primary design layer stratum')}
design_layer_stratum to design_layer_technology (as of_technology)	PATH		338	stratum <= product_definition characterized_product_definition = product_- definition characterized_product_definition characterized_definition = characterized_- product_definition characterized_definition <= property_definition.definition property_definition <= property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship property_definition_relationship.related_- property_definition ->

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object {characterized_object characterized_object.description = 'design layer'} characterized_object => stratum_technology </pre>
DOCUMENTATION_LAYER_- STRATUM	stratum	210	263, 264, 268, 267	<pre> stratum <= product_definition {product_definition product_definition.name = 'documentation layer'} </pre>
documentation_layer_stratum to documentation_layer_technology (as of_technology)	PATH		265, 266, 338	<pre> stratum <= product_definition characterized_product_definition = product_- definition characterized_product_definition product_definition = characterized_- product_definition characterized_definition <- property_definition.definition property_definition <- property_definition.relationship.related_- property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'} property_definition_relationship property_definition_relationship.related_- property_definition -> property_definition property_definition.definition -> </pre>

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_-object {characterized_object characterized_object.description = 'documentation layer'} characterized_object => stratum_technology
LAYER	layer	210		layer <= shape_aspect
name	shape_aspect.name	41	333	layer <= shape_aspect shape_aspect.name
layer to external_definition (as layer_usage)	PATH			layer document_assigned_item = layer document_assigned_item <= applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <= document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'layer definition'}
layer to inter_stratum_extent (as extent)	PATH		340	layer <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_-product_definition

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> characterized_product_definition definition_relationship product_definition_relationship {product_definition_relationship product_definition_relationship.name = 'inter stratum extent'}</pre>
STRATUM	stratum	210	265, 266, 338	<pre> stratum <= product_definition {product_definition (product_definition.name = 'design layer') (product_definition.name = 'documentation layer') (product_definition.name = 'generic stratum')}} stratum <= product_definition product_definition.id (stratum document_assigned_item = stratum document_assigned_item <- applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint {document_usage_constraint document_usage_constraint.subject_element = 'stratum definition'}) (stratum <= product_definition characterized_product_definition = product_- definition</pre>
name	product_definition.id	41		
stratum to external_definition (as stratum_usage)	PATH			

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'stratum usage'} property_definition_relationship.related_ property_definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ object characterized_object => external_definition)
stratum to stratum_technology (as of_technology)	PATH			stratum <= product_definition characterized_product_definition = product_ definition characterized_product_definition characterized_definition = characterized_ product_definition characterized_definition <- property_definition.definition property_definition <- property_definition_relationship.related_ property_definition {property_definition_relationship property_definition_relationship.name = 'technology usage'}

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_relationship property_definition_relationship.relatiing_ property_definition -> property_definition property_definition.definition -> characterized_definition = characterized_ object characterized_object => stratum_technology
STRATUM_AVERAGE_SUR- FACE	stratum_surface	210		{stratum_surface <= shape_aspect shape_aspect.description = 'average surface'} stratum_surface <= shape_aspect shape_aspect.of_shape -> product_definition_shape product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition product_definition product_definition => stratum
stratum_average_surface to stratum (as of_stratum)	PATH			
STRATUM_SURFACE	stratum_surface	210	333, 329	stratum_surface <= shape_aspect stratum_surface stratum_surface <= shape_aspect {shape_aspect
name	PATH			

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'primary surface') (shape_aspect.description = 'secondary surface')}
stratum_surface to stratum (as of_stratum)	PATH		340, 265, 266, 338	shape_aspect.description stratum_surface stratum_surface <= shape_aspect shape_aspect.of_shape -> product_definition_shape product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition product_definition => stratum
STRATUM_TECHNOLOGY	stratum_technology	210		stratum_technology <= characterized_object
laminate_stiffness_class	descriptive_representation_item	45	109	stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation ->

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation { representation.name = 'physical characteristics representation' } representation.items[i] -> { representation_item representation_item.name = 'laminare stiffness class' } representation_item => descriptive_representation_item { descriptive_representation_item (descriptive_representation_item.description = 'fluid like') } (descriptive_representation_item.description = 'conformal coat') (descriptive_representation_item.description = 'stiff laminate') </pre>
name	characterized_object.name	41		<pre> stratum_technology <= characterized_object characterized_object.name </pre>
layer_position	descriptive_representation_item	45		<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation { representation representation.name = 'physical characteristics representation' } </pre>

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] -> {representation_item representation_item.name = 'layer position'} representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'secondary') (descriptive_representation_item.description = 'all') (descriptive_representation_item.description = 'external') (descriptive_representation_item.description = 'primary') (descriptive_representation_item.description = 'internal')}} </pre>
stratum_technology to ee_material (as stratum_material)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- material_designation.definitions[i] material_designation </pre>
stratum_technology to length_data_element (as maximum_feature_size_- requirement)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- </pre>

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum feature size requirement'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as maximum_stratum_thickness)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'maximum thickness'} </pre>

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_stratum_thickness)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum thickness'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_finished_feature_size)	PATH			<pre> stratum_technology <= characterized_object characterized_definition = characterized_- object characterized_definition <- property_definition.definition property_definition <- </pre>

Table 96 – Mapping table for stratum UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature size'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit </pre>
stratum_technology to length_data_element (as minimum_finished_feature_ - spacing)	PATH		109	<pre> stratum_technology <= characterized_object characterized_definition = characterized_ object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'physical characteristics representation'} representation.items[i] -> {representation_item representation_item.name = 'minimum finished feature spacing'} </pre>

Table 96 – Mapping table for stratum UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
stratum_technology to process_specification (as specification)	PATH			stratum_technology <= characterized_object document_assigned_item = characterized_object document_assigned_item <= applied_document_reference.items[i] applied_document_reference applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'process specification'} document => ee_specification

Table 97 – Mapping table for stratum_feature UoF

Application element	AIM element	Source	Rules	Reference path
DESIGN_INTENT_MODIFICATION	shape_modification	210		shape_modification <= shape_aspect
design_intent_modification to assembly_component (as modification_causal_item)	PATH		333, 265, 266, 338	<pre> shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatng_shape_- aspect -> (shape_aspect => component_shape_aspect) (shape_aspect {shape_aspect => (component_terminal) (component_interface_terminal) (laminate_component_interface_terminal) (component_feature) (interconnect_module_component_surface_- feature)}} shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => component_definition)
design_intent_modification to component_external_reference (as modification_causal_item)	PATH			shape_modification <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> {representation representation.name = 'component external reference'} representation
design_intent_modification to component_feature (as modification_causal_item)	PATH		333	shape_modification <= shape_aspect <= shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatng_shape_ aspect -> (shape_aspect => (component_terminal) (component_interface_terminal) (laminatate_component_interface_terminal) (component_feature))

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification to component_feature_external_- reference (as modification_causal_item)	PATH			<pre> shape_modification <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_- representation -> {representation representation.name = 'modification causal feature'} representation representation.items[i] -> {representation_item representation_item.name = 'component feature external reference'} representation_item => descriptive_representation_item </pre>
design_intent_modification to ee_requirement_occurrence (as modification_causal_item)	PATH		333, 340	<pre> shape_modification <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition {property_definition property_definition.name = 'modification causal item'} property_definition <- property_definition_relationship.related_- property_definition </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>{property_definition_relationship property_definition_relationship.name = 'modification_causal_item'} property_definition_relationship.relatin_ property_definition -> property_definition => requirements_property shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatin_shape_- aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature</pre>
design_intent_modification to inter_stratum_feature (as modification_causal_item)	PATH		333	<pre>shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatin_shape_- aspect -> shape_aspect => component_shape_aspect => inter_stratum_feature</pre>
design_intent_modification to laminiate_component (as modification_causal_item)	PATH		333, 340, 263, 264, 268, 267, 265, 266, 338	<pre>shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatin_shape_- aspect -> (shape_aspect => component_shape_aspect =></pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (inter_stratum_feature) (land) (thermal_component_shape_aspect) (fiducial) (probe_access_area) (assembly_group_component_shape_aspect) (integral_shield) (material_removal_component_shape_aspect)) (shape_aspect {shape_aspect.description = 'laminare text string component'} => component_shape_aspect => assembly_group_component_shape_aspect => array_placement_group_component_shape_aspect) (shape_aspect {(shape_aspect.description = 'laminare component') (shape_aspect.description = 'part template occurrence') (shape_aspect.description = 'conductive interconnect element with pre defined transitions') (shape_aspect.description = 'join 2 physical connectivity definition supporting') (shape_aspect.description = 'conductive interconnect element with user defined single transition') (shape_aspect.description = 'unrouted conductive interconnect element') (shape_aspect.description = 'laminare text component') (shape_aspect.description = 'primary stratum indicator symbol') (shape_aspect.description = 'routed interconnect component') </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(shape_aspect.description = 'special symbol laminate component') (shape_aspect.description = 'stratum feature template component'))} (shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition product_definition => {product_definition.description = 'laminated component'}) component_definition)</pre>
design_intent_modification to part_feature (as modification_causal_item)	PATH			<pre>shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship shape_aspect_relationship.related_shape_ aspect -> shape_aspect {([shape_aspect => composite_shape_aspect] [shape_aspect shape_aspect.description = 'part group feature']) (shape_aspect.description = 'part generic feature')}</pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.description = 'polarity indication feature') (shape_aspect.description = 'interconnect module edge segment surface') (shape_aspect.description = 'interconnect module edge surface') (shape_aspect.description = 'interconnect module primary surface') (shape_aspect.description = 'interconnect module secondary surface') (shape_aspect.description = 'interconnect module surface feature') (shape_aspect => primary_orientation_feature) (shape_aspect => secondary_orientation_feature) (shape_aspect => package_body) (shape_aspect => part_tooling_feature) (shape_aspect => thermal_feature) (shape_aspect => part_mounting_feature) (shape_aspect => package_terminal) (shape_aspect => assembly_module_terminal) (shape_aspect => interconnect_module_terminal) (shape_aspect => minimally_defined_bare_die_terminal) (shape_aspect => packaged_part_terminal)

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect => package_body_surface)}
design_intent_modification to stratum (as modification_causal_item)	PATH			shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => stratum
design_intent_modification to stratum_feature (as design_intent)	PATH		333	shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'design intent modification'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification to stratum_feature (as modification_causal_item)	PATH			<pre> shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_feature </pre>
design_intent_modification to stratum_surface (as modification_causal_item)	PATH			<pre> shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'modification causal feature'} shape_aspect_relationship.related_shape_- aspect -> shape_aspect => stratum_surface </pre>
DESIGN_INTENT_MODIFICA- TION_3D_POSITION	mapped_item	43		<pre> {mapped_item <= representation_item representation_item.name = 'positioned design intent modification 3d shape'} </pre>
design_intent_modification_3d_- position to axis_placement_3d (as transformation)	PATH			<pre> mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification_3d_- position to physical_unit_3d_shape (as design_shape)	PATH			<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape'] [representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit[]} representation => shape_representation } </pre>
design_intent_modification_3d_- position to design_intent_modification_non_- planar_2d_shape (as feature_shape)	PATH			<pre> mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation => manifold_surface_shape_representation </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification_3d_- position to design_intent_modification_3d_- shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
DESIGN_INTENT_MODIFI- CATION_PLANAR_POSITION	mapped_item	43		{mapped_item <= representation_item representation_item.name = 'design intent modification planar position'}
design_intent_modification_- planar_position to physical_unit_planar_shape (as design_shape)	PATH			mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition = product_ definition {product_definition product_definition.name = 'interconnect module'}

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => physical_unit}} representation => shape_representation
design_intent_modification_- planar_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d
design_intent_modification_- planar_position to design_intent_modification_- styled_area_representation (as feature_shape)	PATH			mapped_item <= representation_item <= representation.items[i] representation <= property_definition_representation.used_- representation property_definition_representation
design_intent_modification_- planar_position to design_intent_modification_- planar_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <= property_definition_representation.used_- representation property_definition_representation
DESIGN_INTENT_MODIFICATION_SEQUENCE	shape_aspect_relationship	41	125, 126	{shape_aspect_relationship shape_aspect_relationship.name = 'shape modification sequence'}
design_intent_modification_- sequence to design_intent_modification (as precedent_modification)	PATH		333	shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => shape_modification

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
design_intent_modification_- sequence to design_intent_modification (as subsequent_modification)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => shape_modification
ELECTRICAL_ISOLATION_RE- MOVAL_STRUCTURE	shape_modification	210	333	shape_modification <= shape_aspect {shape_aspect shape_aspect.description = 'electrical isolation removal'}
electrical_isolation_removal_- structure to ee_requirement_occurrence (as electrical_isolation_spacing_- requirement)	PATH		340	shape_modification <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition {property_definition property_definition.name = 'spacing requirement'}
				property_definition <= property_definition.relationship.related_ property_definition {property_definition.relationship property_definition.relationship.name = 'spacing requirement'}
				property_definition.relationship.related_ property_definition {property_definition.relationship.name = 'spacing requirement'}
				property_definition.relationship.related_ property_definition <= property_definition <= requirements_property
FILLED_AREA_MATERIAL_- REMOVAL_STRUCTURE	shape_modification	210	333	shape_modification <= shape_aspect {shape_aspect (shape_aspect.description = 'electrical isolation removal') (shape_aspect.description = 'thermal isolation removal')}

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
filled_area_material_removal_- structure to conductive_filled_area (as design_intent)	PATH		333	<pre> shape_modification <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'design intent'} shape_aspect_relationship.relatng_shape_- aspect -> {shape_aspect (shape_aspect.description = 'conductive filled area')} (shape_aspect.description = 'connected filled area')} shape_aspect => stratum_feature </pre>
MATERIAL_ADDITION_FEA- TURE	shape_modification	210		<pre> shape_modification <= shape_aspect {shape_aspect shape_aspect.description = 'material addition feature'} </pre>
material_addition_feature to material_addition_feature_- template (as material_addition_shape)	PATH			<pre> shape_modification<= shape_aspect shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'material addition shape'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect [{shape_aspect shape_aspect.description = 'material addition feature template'}] </pre>

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
MATERIAL_REMOVAL_FEATURE	shape_modification	210	333	shape_modification <= shape_aspect {shape_aspect shape_aspect.description = 'material removal feature'}}
material_removal_feature to material_removal_feature_- template (as material_removal_shape)	PATH			shape_modification <= shape_aspect shape_aspect <= shape_aspect.relationship.related_shape_aspect shape_aspect.relationship {shape_aspect.relationship shape_aspect.relationship.name = 'material removal_shape'} shape_aspect.relationship.related_shape_- aspect -> shape_aspect [{shape_aspect shape_aspect.description = 'material removal feature template'}]
MATERIAL_SHAPE_MODIFICATION	shape_modification	210		shape_modification <= shape_aspect {shape_aspect shape_aspect.description = 'material shape'}
PHYSICAL_COMPONENT_TERMINAL	(component_terminal) (component_interface_terminal)	210 210		(component_terminal <= shape_aspect {shape_aspect (shape_aspect.description = 'interconnect module component stratum based terminal') (shape_aspect.description = 'minimally defined component terminal') }) (component_interface_terminal <= shape_aspect)
STRATUM_FEATURE	stratum_feature	210	333, 322	stratum_feature <= shape_aspect
feature_of_size	descriptive_representation_item	45	342	stratum_feature <= shape_aspect

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation representation.items[i] -> representation_item representation_item.name = 'feature of size' representation_item => descriptive_representation_item {descriptive_representation_item (descriptive_representation_item.description = 'true') (descriptive_representation_item.description = 'false')}
name	shape_aspect.name	41		stratum_feature <= shape_aspect shape_aspect.name
stratum_feature to stratum (as resident_stratum)	PATH		340, 265, 266, 338	stratum_feature <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum
STRATUM_FEATURE_IN - STRATUM_3D_POSITION	mapped_item	43	317, 337, 109	{mapped_item <= representation_item representation_item.name = 'positioned stratum feature shape in stratum'}
stratum_feature_in_stratum_3d_ position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d
stratum_feature_in_stratum_3d_ position to stratum_3d_shape (as stratum_shape)	PATH		340, 265, 266, 338	mapped_item <= representation_item <= representation.items[i] {[representation representation.name = '3d bound volume shape'] representation <= property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition => stratum[] representation => shape_representation
stratum_feature_in_stratum_3d - position to stratum_feature_non_planar_2d - shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> {representation representation.name = 'stratum feature non planar 2d shape'} representation => shape_representation => manifold_surface_shape_representation
stratum_feature_in_stratum_3d - position to stratum_feature_3d_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
STRATUM_FEATURE_IN - STRATUM_PLANAR_POSI- TION	mapped_item	43	109, 318	{mapped_item <= representation_item representation_item.name = 'positioned stratum feature shape in stratum'}
stratum_feature_in_stratum - planar_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d

Table 97 – Mapping table for stratum_feature UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_feature_in_stratum_- planar_position to stratum_feature_planar_shape (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
stratum_feature_in_stratum_- planar_position to stratum_feature_styled_area_- representation (as feature_shape)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- property_definition_representation.used_ representation property_definition_representation
stratum_feature_in_stratum_- planar_position to stratum_planar_shape (as stratum_shape)	PATH		340, 265, 266, 338	mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape'] [representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition

Table 97 – Mapping table for stratum_feature UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition characterized_product_definition = product_ definition product_definition => stratum[] representation => shape_representation
THERMAL_ISOLATION_RE- MOVAL_STRUCTURE	shape_modification	210	333	shape_modification <= shape_aspect {shape_aspect shape_aspect.description = 'thermal isolation removal'}

Table 98 – Mapping table for stratum_position UoF

Application element	AIM element	Source	Rules	Reference path
ADJACENT_STRATUM_SURFACE_DEFINITION	shape_aspect_relationship	41	2, 3	shape_aspect_relationship {shape_aspect_relationship.description = 'adjacent stratum surface definition'}
name	shape_aspect_relationship.name	41		
adjacent_stratum_surface_- definition to stratum_surface (as precedent_surface)	PATH		333	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'secondary surface'} shape_aspect => stratum_surface
adjacent_stratum_surface_- definition to stratum_surface (as subsequent_surface)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect {shape_aspect shape_aspect.description = 'primary surface'} shape_aspect => stratum_surface
ADJACENT_STRATUM_SURFACE_EMBEDDED_COMPONENT_SURFACE_DEFINITION	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'adjacent stratum surface embedded component surface'}
name	descriptive_representation_- item.description	45		shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition {property_definition property_definition.description = 'adjacent stratum surface embedded component surface characterization'} property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_ representation -> representation {representation representation.name = 'adjacent stratum surface embedded component surface name'} representation.items[i] -> representation_item => descriptive_representation_item descriptive_representation_item.description
shape_element_relationship to stratum_surface (as precedent_item)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect => stratum_surface
shape_element_relationship to stratum_surface (as subsequent_item)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_surface
shape_element_relationship to component_feature (as precedent_item)	PATH			shape_aspect_relationship shape_aspect_relationship.relatng_shape_ aspect -> shape_aspect {(shape_aspect => component_terminal)}

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => component_feature) {(shape_aspect => part_mounting_feature => bare_die_surface => bare_die_top_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_edge_surface) (shape_aspect => bare_die_edge_surface) (shape_aspect => bare_die_edge_segment_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_bottom_surface) (shape_aspect => package_body_surface => package_body_top_surface) (shape_aspect => package_body_surface => package_body_edge_surface) (shape_aspect => package_body_edge_segment_surface) (shape_aspect => package_body_surface => package_body_bottom_surface)} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {(shape_aspect => component_terminal) </pre>
shape_element_relationship to component_feature (as subsequent_item)	PATH			

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> (shape_aspect => component_feature) {(shape_aspect => part_mounting_feature => bare_die_surface => bare_die_top_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_edge_surface) (shape_aspect => bare_die_edge_surface) (shape_aspect => bare_die_edge_segment_surface) (shape_aspect => part_mounting_feature => bare_die_surface => bare_die_bottom_surface) (shape_aspect => package_body_surface => package_body_top_surface) (shape_aspect => package_body_surface => package_body_edge_surface) (shape_aspect => package_body_edge_segment_surface) (shape_aspect => package_body_surface => package_body_bottom_surface)} </pre>
INTERCONNECT_MODULE - STRATUM_ASSEMBLY_RELATIONSHIP	assembly_component_usage	44	192, 354, 339, 193	<pre> {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} </pre>

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
reference_designation	assembly_component_- usage.reference_designator	44		
interconnect_module_stratum_- assembly_relationship to interconnect_module (as assembly)	PATH		263, 264, 268, 267, 265, 266, 338	assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> {product_definition [product_definition.name = 'interconnect module'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical design']}] product_definition => physical_unit
interconnect_module_stratum_- assembly_relationship to stratum (as component)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.related_- product_definition -> product_definition => stratum
STRATUM_INTERCONNECT_- MODULE_3D_POSITION	mapped_item	43	337, 323, 109	{mapped_item <= representation_item representation_item.name = '3d positioned stratum shape in interconnect module'}
stratum_interconnect_module_- 3d_position to axis_placement_3d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_3d

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
stratum_interconnect_module_- 3d_position to interconnect_module_stratum_- assembly_relationship (as stratum_component)	PATH		340, 339, 354	<pre> mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => placement => axis2_placement_3d} representation_item <- representation.items[i] representation <- property_definition_representation.used_- representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition_relationship {product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} product_definition_relationship => product_definition_usage => assembly_component_usage </pre>
stratum_interconnect_module_- 3d_position to physical_unit_3d_shape (as interconnect_module_shape)	PATH		340, 265, 266, 263, 264, 268, 267	<pre> mapped_item <= representation_item <- representation.items[i] {[representation representation.name = '3d bound volume shape'] representation <- property_definition_representation.used_- representation </pre>

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit}} representation => shape_representation </pre>
stratum_interconnect_module_- 3d_position to stratum_3d_shape (as stratum_shape)	PATH		340, 265, 266, 338	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin']} [representation_item => geometric_representation_item => axis2_placement_3d]} representation_map.mapped_representation -> {[representation representation.name = '3d bound volume shape']} [representation <- </pre>

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_ - representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ - product_definition characterized_product_definition definition product_definition => stratum[]} representation => shape_representation
STRATUM_INTERCONNECT_- MODULE_PLANAR_POSITION	mapped_item	43	109, 324	{mapped_item <= representation_item representation_item.name = 'planar positioned stratum shape in interconnect module'}
stratum_interconnect_module_- planar_position to axis_placement_2d (as transformation)	PATH			mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => placement => axis2_placement_2d
stratum_interconnect_module_- planar_position to interconnect_module_stratum_- assembly_relationship (as stratum_component)	PATH		340, 354	mapped_item mapped_item.mapping_target -> representation_item {representation_item => geometric_representation_item => placement => axis2_placement_2d} representation_item <-

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> representation.items[i] representation <- property_definition_representation.used_ representation property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_ product_definition characterized_product_definition characterized_product_definition = product_ definition_relationship {product_definition_relationship product_definition_relationship.name = 'interconnect module stratum assembly relationship'} product_definition_relationship => product_definition_usage => assembly_component_usage mapped_item <= representation_item <- representation.items[i] {[representation representation.name = 'planar projected shape']} [representation <- property_definition_representation.used_ representation property_definition_representation </pre>
stratum_interconnect_module_- planar_position to physical_unit_planar_shape (as interconnect_module_shape)	PATH		340, 265, 266, 338, 263, 264, 268, 267	

Table 98 – Mapping table for stratum_position UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre> property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition {product_definition product_definition.name = 'interconnect module'} product_definition => physical_unit[] representation => shape_representation </pre>
stratum_interconnect_module_- planar_position to stratum_planar_shape (as stratum_shape)	PATH		340, 265, 266, 338	<pre> mapped_item mapped_item.mapping_source -> representation_map {representation_map representation_map.mapping_origin -> representation_item [representation_item.name = 'origin'] [representation_item => geometric_representation_item => axis2_placement_2d]} representation_map.mapped_representation -> {representation representation.name = 'planar projected shape'} [representation <- property_definition_representation.used_- representation </pre>

Table 98 – Mapping table for stratum_position UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				<pre>property_definition_representation property_definition_representation.definition -> {property_definition => product_definition_shape} property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition product_definition => stratum[]} representation => shape_representation</pre>

Table 99 – Mapping table for styled_curve UoF

Application element	AIM element	Source	Rules	Reference path
CURVE_STYLE	(curve_style) (representation)	46 43	68, 71	(curve_style) (representation [representation.context_of_items -> representation_context [representation_context representation_context => global_unit_assigned_context] [representation_context representation_context => global_uncertainty_assigned_context] [representation_context representation_context => parametric_representation_context]]) [representation representation.context_of_items -> representation_context representation_context.context_type representation_context.context_type = 'curve style parametric context'] [representation representation.name = 'curve style parameters']])
corner_style	PATH			representation representation.items[i] -> representation_item => descriptive_representation_item {descriptive_representation_item {descriptive_representation_item.name = 'corner_style'}}

Table 99 – Mapping table for styled_curve UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				<pre>(descriptive_representation_item.description = 'chord 2 extend') (descriptive_representation_item.description = 'extend') (descriptive_representation_item.description = 'round') (descriptive_representation_item.description = 'truncate')}</pre>
end_extension	PATH			<pre>representation representation.items[i] -> representation_item => descriptive_representation_item {descriptive_representation_item {descriptive_representation_item.name = 'end extension'} (descriptive_representation_item.description = 'extend') (descriptive_representation_item.description = 'truncate')}</pre>
end_style	PATH			<pre>representation representation.items[i] -> representation_item => descriptive_representation_item {descriptive_representation_item {descriptive_representation_item.name = 'end style'} (descriptive_representation_item.description = 'round') (descriptive_representation_item.description = 'square')}</pre>
name	descriptive_representation_ - item.description	43		<pre>representation representation.items[i] -> representation_item => descriptive_representation_item {descriptive_representation_item</pre>

Table 99 – Mapping table for styled_curve UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				descriptive_representation_item.name = 'curve style name' descriptive_representation_item.description
curve_style to curve_style_font (as curve_font)	PATH			curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select curve_style_font_select = curve_style_font curve_style_font
curve_style to curve_style_font_and_scaling (as curve_font)	PATH			curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_and_scaling curve_style_font_and_scaling
curve_style to externally_defined_curve_font (as curve_font)	PATH			curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select curve_style_font_select = externally_defined_- curve_font externally_defined_curve_font
curve_style to length_data_element (as curve_width)	PATH			representation representation.items[i] -> representation_item {representation_item representation_item.name = 'curve width'} representation_item => measure_representation_item <= measure_with_unit => length_measure_with_unit
curve_style to length_data_element (as width_uncertainty)	PATH			representation representation.context_of_items -> representation_context representation_context => global_uncertainty_assigned_context

Table 99 – Mapping table for styled_curve UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				global_uncertainty_assigned_- context.uncertainty[i] -> uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit
curve_style_to_solid_curve (as curve_font)	PATH			curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select curve_style_font_select = pre_defined_curve_- font pre_defined_curve_font => solid_curve_font
CURVE_STYLE_FONT	curve_style_font	46		
CURVE_STYLE_FONT_AND_- SCALING	curve_style_font_and_scaling	46	69	
curve_font_scaling	curve_style_font_and_- scaling_curve_font_scaling	46		
name	curve_style_font_and_- scaling.name	46		
curve_style_font_and_scaling to curve_style_font (as curve_font)	PATH			curve_style_font_and_scaling curve_style_font_and_scaling.curve_font -> curve_style_font_select curve_style_font_select = curve_style_font curve_style_font
curve_style_font_and_scaling to externally_defined_curve_font (as curve_font)	PATH			curve_style_font_and_scaling curve_style_font_and_scaling.curve_font -> curve_style_font_select curve_style_font_select = externally_defined_- curve_font externally_defined_curve_font

Table 99 – Mapping table for styled_curve UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
curve_style_font_and_scaling to solid_curve (as curve_font)	PATH			curve_style_font_and_scaling curve_style_font_and_scaling.curve_font -> curve_style_font_select curve_style_font_select = pre_defined_curve_- font pre_defined_curve_font => solid_curve_font
EXTERNALLY_DEFINED_- CURVE_FONT	externally_defined_curve_font	46		
SOLID_CURVE	solid_curve_font	210		solid_curve_font <= pre_defined_curve_font
STYLED_CURVE_OCCUR- RENCE	annotation_curve_occurrence	46		
styled_curve_occurrence to curve_style (as of_curve_style)	PATH			annotation_curve_occurrence <= annotation_occurrence <= styled_item styled_item.styles[i] -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = curve_style curve_style
styled_curve_occurrence to curve (as styled_curve)	PATH			annotation_curve_occurrence <= annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => curve

Table 100 – Mapping table for styled_text UoF

Application element	AIM element	Source	Rules	Reference path
FONT_DEFINITION	externally_defined_text_font	46		
REGISTERED_FONT	externally_defined_text_font	46	281, 282	
font_code	externally_defined_item.item_id	41		externally_defined_text_font <= externally_defined_item externally_defined_item.item_id
registered_font to ee_document (as registration_document)	PATH			externally_defined_text_font <= externally_defined_item document_assigned_item = externally_defined_ item document_assigned_item <= applied_document_reference.items[i] applied_document_reference <= document_reference document_reference.assigned_document -> {document document.kind -> document_type document_type.product_data_type = 'font registration document'} (document) (document => ee_specification)
STYLED_TEXT	text_literal	46	356	
styled_text to font_definition (as text_font)	PATH		356	text_literal text_literal.font -> font_select font_select = externally_defined_text_font externally_defined_text_font

Table 100 – Mapping table for styled_text UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
styled_text to registered_font (as text_font)	PATH		356	text_literal text_literal.font -> font_select font_select = externally_defined_text_font externally_defined_text_font

Table 101 – Mapping table for test_probe UoF

Application element	AIM element	Source	Rules	Reference path
INTERNAL_PROBE_ACCESS_-AREA	probe_access_area	210	340, 338, 268	<pre>probe_access_area <= component_shape_aspect <= shape_aspect {shape_aspect [shape_aspect.description = 'internal probe access area'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminated component'] [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'layout occurrence']]}</pre>
internal_probe_access_area to stratum_feature (as stratum_feature_implementation)	PATH		333	<pre>probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship</pre>

Table 101 – Mapping table for test_probe UoF (Continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect_relationship shape_aspect_relationship.name = 'implementation'} shape_aspect_relationship.relatng_shape_- aspect -> shape_aspect => stratum_feature
PROBE_ACCESS_AREA	probe_access_area	210	333, 340, 338, 268	probe_access_area <= component_shape_aspect <= shape_aspect {[(shape_aspect.description = 'probe access area') (shape_aspect.description = 'internal probe access area')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_- product_definition characterized_product_definition characterized_product_definition = product_- definition [product_definition => component_definition] [product_definition product_definition.description = 'laminare component']]}
probe_access_area to connection_zone (as probe_contact_zone)	PATH		333	probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship

Table 101 – Mapping table for test_probe UoF (Concluded)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect_relationship shape_aspect_relationship.name = 'probe contact zone'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'connection zone'}
probe_access_area to stratum_feature (as probed_layout_item)	PATH			probe_access_area <= component_shape_aspect <= shape_aspect <- shape_aspect_relationship.relatng_shape_- aspect shape_aspect_relationship {shape_aspect_relationship shape_aspect_relationship.name = 'probed layout item'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => stratum_feature

Table 102 – Mapping table for wireframe_2d_model_with_topology UoF

Application element	AIM element	Source	Rules	Reference path
EDGE_BASED_2D_WIRE-FRAME	edge_based_2d_wireframe_- shape_representation	210		edge_based_2d_wireframe_shape_representation <= shape_representation
SHELL_BASED_2D_WIRE-FRAME	shell_based_2d_wireframe_- shape_representation	210		shell_based_2d_wireframe_shape_representation <= shape_representation {shape_representation <= representation.context_of_items -> representation.context => geometric_representation.context.coordinate_- space_dimension = 2}

Table 103 – Mapping table for wireframe_with_topology UoF

Application element	AIM element	Source	Rules	Reference path
EDGE_BASED_WIREFRAME	edge_based_wireframe_shape_- representation	501		edge_based_wireframe_shape_representation <= shape_representation
SHELL_BASED_WIREFRAME	shell_based_wireframe_shape_- representation	502		shell_based_wireframe_shape_representation <= shape_representation

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and contains the types, entity specializations, rules, and functions that are specific to this Part of ISO 10303.

EXPRESS specification:

```

*)
SCHEMA electronic_assembly_interconnect_and_packaging_design;

REFERENCE FROM support_resource_schema -- ISO 10303-41
(label);

USE FROM action_schema -- ISO 10303-41
(action,
action_directive,
action_method,
action_method_relationship,
action_relationship,
action_request_solution,
action_request_status,
action_status,
directed_action,
executed_action,
versioned_action_request);

USE FROM aic_advanced_brep; -- ISO 10303-514

USE FROM aic_csg; -- ISO 10303-515

USE FROM aic_edge_based_wireframe; -- ISO 10303-501

USE FROM aic_geometrically_bounded_2d_wireframe; -- ISO 10303-503

USE FROM aic_geometrically_bounded_surface; -- ISO 10303-507

USE FROM aic_geometrically_bounded_wireframe; -- ISO 10303-510

USE FROM aic_manifold_surface; -- ISO 10303-509

USE FROM aic_manifold_subsurface; -- ISO 10303-521

USE FROM aic_shell_based_wireframe; -- ISO 10303-502

USE FROM application_context_schema -- ISO 10303-41
(application_context,
application_context_element,
application_protocol_definition,
product_concept_context,
product_context,

```

```
    product_definition_context);

USE FROM approval_schema -- ISO 10303-41
    (approval,
     approval_date_time,
     approval_person_organization,
     approval_relationship,
     approval_role,
     approval_status);

USE FROM certification_schema -- ISO 10303-41
    (certification,
     certification_type);

USE FROM configuration_management_schema -- ISO 10303-44
    (configuration_design,
     configuration_effectivity,
     configuration_item);

USE FROM contract_schema -- ISO 10303-41
    (contract,
     contract_type);

REFERENCE FROM date_time_schema -- ISO 10303-41
    (leap_year);

USE FROM date_time_schema -- ISO 10303-41
    (calendar_date,
     date_and_time,
     date_time_role,
     date_role,
     date);

USE FROM document_schema -- ISO 10303-41
    (document,
     document_relationship,
     document_type,
     document_usage_constraint,
     document_with_class);

USE FROM effectivity_schema -- ISO 10303-41
    (effectivity,
     dated_effectivity,
     lot_effectivity,
     serial_numbered_effectivity);

USE FROM external_reference_schema -- ISO 10303-41
    (externally_defined_item,
     external_source,
     external_source_relationship,
     pre_defined_item);
```



```

USE FROM geometric_model_schema -- ISO 10303-42
(boolean_operand,
 boolean_operator,
 boolean_result,
 circular_area,
 csg_primitive,
 csg_select,
 elliptic_area,
 extruded_area_solid,
 half_space_2d,
 polygonal_area,
 primitive_2d,
 rectangle_domain,
 rectangled_half_space,
 rectangular_area,
 revolved_area_solid);

```

```

REFERENCE FROM geometry_schema -- ISO 10303-42
(dummy_gri);

```

```

USE FROM geometry_schema -- ISO 10303-42
(axis2_placement_2d,
 axis2_placement_3d,
 bounded_curve,
 b_spline_curve,
 cartesian_point,
 cartesian_transformation_operator,
 cartesian_transformation_operator_2d,
 circle,
 composite_curve,
 composite_curve_segment,
 curve,
 curve_bounded_surface,
 ellipse,
 geometric_representation_context,
 geometric_representation_item,
 hyperbola,
 line,
 outer_boundary_curve,
 parabola,
 placement,
 plane,
 polyline,
 trimmed_curve);

```

```

USE FROM group_schema -- ISO 10303-41
(group,
 group_relationship);

```

```

USE FROM management_resources_schema -- ISO 10303-41
(action_assignment,
 action_request_assignment,

```

```

approval_assignment,
certification_assignment,
contract_assignment,
date_assignment,
date_and_time_assignment,
document_reference,
group_assignment,
organization_assignment,
person_assignment,
person_and_organization_assignment,
security_classification_assignment);

```

```

USE FROM material_property_definition_schema -- ISO 10303-45
(material_designation,
material_designation_characterization,
material_property,
product_material_composition_relationship,
property_definition_relationship);

```

```

USE FROM material_property_representation_schema -- ISO 10303-45
(data_environment,
material_property_representation);

```

```

USE FROM measure_schema -- ISO 10303-41
(amount_of_substance_measure,
amount_of_substance_measure_with_unit,
amount_of_substance_unit,
area_measure,
area_measure_with_unit,
area_unit,
celsius_temperature_measure_with_unit,
context_dependent_measure,
context_dependent_unit,
conversion_based_unit,
count_measure,
derived_unit,
descriptive_measure,
electric_current_measure,
electric_current_measure_with_unit,
electric_current_unit,
global_unit_assigned_context,
length_measure_with_unit,
length_unit,
luminous_intensity_measure,
luminous_intensity_measure_with_unit,
luminous_intensity_unit,
mass_measure,
mass_measure_with_unit,
mass_unit,
measure_value,
measure_with_unit,
named_unit,

```

```

numeric_measure,
parameter_value,
plane_angle_measure_with_unit,
plane_angle_unit,
positive_plane_angle_measure,
ratio_measure_with_unit,
ratio_unit,
si_unit,
solid_angle_measure,
solid_angle_measure_with_unit,
solid_angle_unit,
thermodynamic_temperature_measure,
thermodynamic_temperature_measure_with_unit,
thermodynamic_temperature_unit,
time_measure,
time_measure_with_unit,
time_unit,
volume_measure,
volume_measure_with_unit,
volume_unit);

```

USE FROM person_organization_schema -- ISO 10303-41

```

(address,
organization,
organization_relationship,
organization_role,
organizational_address,
organizational_project,
person,
person_and_organization,
person_and_organization_role,
person_role,
personal_address);

```

USE FROM presentation_appearance_schema -- ISO 10303-46

```

(box_height,
box_rotate_angle,
box_slant_angle,
box_width,
character_glyph_style_outline,
character_glyph_style_stroke,
curve_style,
curve_style_font,
curve_style_font_and_scaling,
curve_style_with_ends_and_corners,
curve_style_with_extension,
externally_defined_curve_font,
externally_defined_hatch_style,
externally_defined_tile,
externally_defined_tile_style,
fill_area_style,
fill_area_style_colour,

```

```

fill_area_style_hatching,
fill_area_style_tile_coloured_region,
fill_area_style_tile_curve_with_style,
fill_area_style_tile_symbol_with_style,
fill_area_style_tiles,
one_direction_repeat_factor,
pre_defined_curve_font,
presentation_style_by_context,
styled_item,
text_style,
text_style_for_defined_font,
two_direction_repeat_factor);

```

```
USE FROM presentation_definition_schema -- ISO 10303-46
```

```

(annotation_curve_occurrence,
annotation_fill_area,
annotation_fill_area_occurrence,
annotation_occurrence,
annotation_symbol,
annotation_text,
annotation_text_character,
annotation_text_occurrence,
externally_defined_symbol,
symbol_representation,
symbol_representation_map,
symbol_target,
text_literal,
text_literal_with_associated_curves,
text_literal_with_extent,
text_string_representation);

```

```
USE FROM presentation_organization_schema -- ISO 10303-46
```

```
(presentation_representation);
```

```
USE FROM presentation_resource_schema -- ISO 10303-46
```

```

(character_glyph_font_usage,
character_glyph_symbol_outline,
colour,
colour_rgb,
colour_specification,
externally_defined_text_font,
text_font_family,
text_font_in_family);

```

```
USE FROM product_concept_schema -- ISO 10303-44
```

```
(product_concept);
```

```
USE FROM product_definition_schema -- ISO 10303-41
```

```

(product,
product_category,
product_category_relationship,
product_definition,

```

```

product_definition_formation,
product_definition_formation_relationship,
product_definition_formation_with_specified_source,
product_definition_relationship,
product_definition_with_associated_documents,
product_related_product_category);

```

```

USE FROM product_property_definition_schema -- ISO 10303-41
(characterized_object,
characterized_product_definition,
product_definition_shape,
property_definition,
shape_aspect,
shape_aspect_relationship);

```

```

REFERENCE FROM product_property_definition_schema -- ISO 10303-41
(acyclic_shape_aspect_relationship);

```

```

USE FROM product_property_representation_schema -- ISO 10303-41
(context_dependent_shape_representation,
property_definition_representation,
shape_definition_representation,
shape_representation,
shape_representation_relationship);

```

```

USE FROM product_structure_schema -- ISO 10303-44
(alternate_product_relationship,
assembly_component_usage,
assembly_component_usage_substitute,
make_from_usage_option,
next_assembly_usage_occurrence,
product_definition_usage,
promissory_usage_occurrence,
quantified_assembly_component_usage,
specified_higher_usage_occurrence);

```

```

USE FROM qualified_measure_schema -- ISO 10303-45
(descriptive_representation_item,
expanded_uncertainty,
measure_qualification,
measure_representation_item,
precision_qualifier,
type_qualifier,
qualitative_uncertainty);

```

```

USE FROM representation_schema -- ISO 10303-43
(compound_item_definition,
compound_representation_item,
global_uncertainty_assigned_context,
item_defined_transformation,
list_representation_item,
mapped_item,

```

```

parametric_representation_context,
representation,
representation_context,
representation_item,
representation_relationship,
representation_relationship_with_transformation,
set_representation_item,
uncertainty_measure_with_unit);

```

```

USE FROM security_classification_schema -- ISO 10303-41
(security_classification,
security_classification_level);

```

```

USE FROM shape_aspect_definition_schema -- ISO 10303-47
(apex,
centre_of_symmetry,
composite_shape_aspect,
derived_shape_aspect,
extension,
geometric_alignment,
geometric_intersection,
shape_aspect_deriving_relationship,
symmetric_shape_aspect,
tangent);

```

```

USE FROM shape_dimension_schema -- ISO 10303-47
(angular_location,
angular_size,
dimensional_characteristic_representation,
dimensional_location,
dimensional_location_with_path,
dimensional_size,
shape_dimension_representation);

```

```

USE FROM shape_tolerance_schema -- ISO 10303-47
(dimension_related_tolerance_zone_element,
geometric_tolerance,
geometric_tolerance_relationship,
modified_geometric_tolerance,
plus_minus_tolerance,
projected_zone_definition,
runout_zone_definition,
runout_zone_orientation,
runout_zone_orientation_reference_direction,
tolerance_value,
tolerance_zone,
tolerance_zone_definition,
tolerance_zone_form);

```

```

REFERENCE FROM topology_schema -- ISO 10303-42
(dummy_tri);

```

```
USE FROM topology_schema -- ISO 10303-42
(edge_curve,
 edge_loop,
 face,
 face_surface);
```

```
( *
```

NOTE The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
aic_advanced_brep	ISO 10303-514
aic_csg	ISO 10303-515
aic_edge_based_wireframe	ISO 10303-501
aic_geometrically_bounded_2d_wireframe	ISO 10303-503
aic_geometrically_bounded_surface	ISO 10303-507
aic_shell_based_wireframe	ISO 10303-502
application_context_schema	ISO 10303-41
approval_schema	ISO 10303-41
certification_schema	ISO 10303-41
configuration_management_schema	ISO 10303-44
contract_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41
effectivity_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42

geometry_schema	ISO 10303-42
group_schema	ISO 10303-41
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45
material_property_representation_schema	ISO 10303-45
measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
presentation_appearance_schema	ISO 10303-46
presentation_definition_schema	ISO 10303-46
presentation_organization_schema	ISO 10303-46
presentation_resource_schema	ISO 10303-46
product_concept_schema	ISO 10303-44
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41
shape_aspect_definition_schema	ISO 10303-47
shape_dimension_schema	ISO 10303-47
shape_tolerance_schema	ISO 10303-47
topology_schema	ISO 10303-42

5.2.1 Fundamental concepts and assumptions

This part of ISO 10303 is designed to be used in the concurrent engineering of modular electronic products. A fundamental concept of this schema is that the organization controls the configuration of product models for these types of product designs. This schema is not designed to control product designs through drawing control. It is designed to provide a vehicle to control the configuration of the design and associated data. The design in this context includes the requirements and constraints exchanged between the design organization, data suppliers (e.g., component suppliers, value added suppliers), product suppliers (e.g., piece part vendors, substrate vendors), and the customers or marketing representatives for those customers. Several engineering disciplines may share data using this schema (e.g., system, mechanical, electrical, manufacturing, inspection, logistics) for product definition and product analysis purposes. This schema provides requirements definition using requirements property definition, decomposition and operations on property definition. This schema provides a functional definition using externally defined models or the functional decomposition model defined in this schema. This schema provides a physical definition for packages, piece parts, interconnect substrate, bare die, mechanical parts, and assemblies. The physical definition includes a geometric dimensioning and tolerancing model that may be applied to any physical model. This schema provides design and usage views for functional and physical models to support design abstraction and intellectual property aspects. This schema provides data suppliers a standard infrastructure with which to build explicit product data models for modular electronic products that may be incorporated into a piece part or technology library. The explicit product data models may be standardized as instances of this schema. This schema provides support for a design organization to archive data using this schema as a design view in one product category and the marketing organization to specify the usage view in a more generic category. The functional usage view in this schema includes an explicit interface definition to allow integration with external simulation environments.

The support for this includes the capability to define product terminals only in a usage view. The product models in this schema include both abstract functional product model and physical product models. Inclusion of the functional model provides the ability to categorize functional models using the product_category entity. This schema provides support for multiple functional definitions to exist in a repository for a particular usage view, but the specific definition desired is included in the functional model exchanged. Parameters (e.g., resistance) whose values are fundamental properties of the materials and their composition and therefore independent of the product version may be exchanged using the subtypes of product_category in this schema. Parameters (e.g., timing related) that are dependent on the version of the product may be exchanged using characteristics associated with a particular product_definition entity. Signal properties and composition may be exchanged using external_definition for the interface to industry domain specific name definitions.

5.2.1.1 Relating the shape of a product to its configuration data

The shape of products in this part of ISO 10303 is represented by the shape_representation entity. This entity and its subtypes define the geometric or topological entities which make up a particular representation type. Each product or assembly that is of interest will be given by an instance of the product entity. Each product will in turn have at least one version given by an instance of the product_definition_entity entity. Each version may have one or more definitions given by the product_definition

entity. Each definition may have its shape represented. This is done using the AIM entities by relating the product_definition instance to the appropriate shape_representation instance. The concept of the shape of a product_definition is given by an instance of the entity product_definition_shape. That shape of the product is then connected to the shape_representation entity by an instance of shape_definition_representation.

A shape_definition_representation inherits attributes from its supertype property_definition_representation that reference the shape_representation entity that contains the geometry or topology for the shape of the product and a select type called characterized_definition. This type allows for the representation of a shape_aspect, shape_aspect_relationship, characterized_object, or a characterized_product_definition which is also a select type. To specify that the property of interest is the shape of the product_definition the product_definition_shape subtype of property_definition shall be used. The product_definition_shape entity is constrained to use the characterized_product_definition select type. The characterized_product_definition select type allows for the representation of a product_definition or a product_definition_relationship via a reference through its select list. This reference ensures that the product_definition_shape entity will be used to define the shape of a product_definition. If the property of interest is an aspect of the shape of the product_definition or the relationship between two aspects of the shape of the product_definition the property_definition shall be used and its definition attribute shall reference an instance of shape_aspect or shape_aspect_relationship. In order to specify the shape of a product the product_definition entity will be referenced here.

5.2.1.2 Relating the shape of a component to the shape of its assembly

There are two methods that may be used to relate the shape of a component to the shape of the assembly in which it is assembled. The first method consists of defining the shape for each component and assembly, and then relating the two shapes and providing the information that defines the orientation of the component with respect to the assembly through a transformation. The second method consists of defining the shape for each component and assembly, and then incorporating the shape of the component directly in the shape of the assembly. The first method shall be used to relate the shapes that are represented by different representation types.

The second method may be used for the incorporation of the representation of a component into the representation of an assembly if the two types are the same. Both methods employ the `shape_representation` and `product_definition` entities. The first method also employs the `product_definition_relationship`, `shape_representation_relationship` AND `representation_relationship_with_transformation` (i.e., an instance of each entity forming a complex entity instance of the two entities with an AND relationship) and `context_dependent_shape_representation` entities.

The second method employs the `mapped_item` and `representation_map` entities. When using the first method to relate the shape of the component to the shape of the assembly, each of the `shape_representation` entities that define the shapes of the component and assembly product definitions is related through references in the `shape_representation_relationship` entity. Orientation information, in this case, will be provided by the formation of a complex instance of the `shape_representation_relationship` AND `representation_relationship_with_transformation` entities. The `representation_relationship_with_transformation` entity references a transformation which is a select type allowing the orientation to be defined using an `axis2_placement_3d` entity in each representation for an `item_defined_transformation` or a `cartesian_transformation_operator` entity for a `functionally_defined_transformation`. In addition, an instance of the `context_dependent_shape_representation` must be given to explicitly relate the `shape_representation_relationship` that defines the relationship of the two shapes to the `product_definition_relationship` that defines the assembly-component relationship between the two product definitions.

When using the second method to relate the shape of the component to the shape of the assembly, the `shape_representation` entity that defines the geometry or topology for the shape of the component product is referenced by an instance of a `representation_map` entity that is referenced by the `mapping_source` attribute of an instance of the `mapped_item` entity. The attribute `mapped_representation` of the `representation_map` will reference the `shape_representation` subtype that defines the representation of the shape. The instance of the `mapped_item` entity is then added to the set of items in the `shape_representation` entity that defines the geometry or topology for the assembly product.

5.2.1.3 Types of shape representation

This part of ISO 10303 defines several types of representation for shapes of items - both 2d and 3d wireframe representations using edge based and shell based models, 2d wireframe representations that are geometrically bounded, surface models that are geometrically bounded, CSG solid models, advanced boundary representation solid models, extruded solid models, 2d Curves and basic 2d Curves. Each of the 3d types of representation is self contained meaning that one type may not contain another type. Each of the types is given by a subtype of the shape_representation entity. Each subtype contains local rules that govern the types of geometric or topological entities that can be used in it. Every shape_representation must be one of the subtypes unless it is used as the representation of the shape of an assembly used in method one above.

In that case, the shape_representation instance will contain only axis2_placement_3d entities in its set of items in order to define the orientation of the components' representations in it. Since the rules in each of the subtypes will conflict with each other, any shape_representation that is referenced by a representation_map to implement method two above must be of the same type as the shape_representation that has the mapped_item that references that representation_map in its set of items.

5.2.1.4 Use of global rules

Many of the relationships among different entities in the integrated resource parts of ISO 10303 are specified using the most generic cardinality of zero or more between two related entities. This cardinality means that the relationship is optional or there may be one or more instances of a related entity that is related to a single instance of the relating entity. This part of ISO 10303 uses global rules to constrain that cardinality. In some cases the constraint is made to be one to one, \and in some instances at least one.

EXAMPLE 1 Examples of these rules include contract_requires_person_organization, approval_requires_approval_date_time, and certification_requires_approval for exactly one, and change_request_requires_person_organization and product_requires_product_version for one or more.

Global rules are also used to restrict the values of STRING type attributes to be only those that are applicable within the context of electronic assembly, interconnect and packaging design.

EXAMPLE 2 Examples of these rules are restrict_approval_status and restrict_person_organization_role.

Global rules are also used to collect in one location all the constraints required to implement a given ARM concept.

EXAMPLE 3 Examples of these rules include operational_requirement_occurrence_relationship_constraint, geometric_tolerance_qualifier_constraint, and feature_group_relationship_constraint.

5.2.1.5 Assignment of units

Units are assigned to the representation of shape on a global basis. This is done by the creation of an instance of the `global_unit_assigned_context`. This entity contains an attribute that allows for a set of units to be assigned to a `representation_context`. Each of the `shape_representation` entities has a context for its representation. If units of measure are desired for a particular instance of `shape_representation` then that instance shall have a `global_unit_assigned_context` in its `context_of_items` attribute.

5.2.1.6 Terminal definition

Terminals are defined as aspects of the shape of a product in this schema. This is true for both terminals of functional products and terminals of physical products. The properties of the terminals interconnected in a network defined by the `functional_network_definition` may be evaluated by experimental means to determine suitability of the `product_definition`. The geometric representation of the functional product terminal is the schematic symbol for the terminal. The geometric representation of the functional product shape is the schematic symbol for the functional product.

NOTE This edition of this part of ISO 10303 does not standardize the exchange of presentation of geometric representations of terminals of functional products or shapes of functional products. In this edition of the standard, there is no standardized way to specify the position of the schematic terminal symbol in the schematic symbol for the functional product. This structure is included to assure interoperability with other parts of ISO 10303.

5.2.1.7 Product instances

This schema separates the "part" in the "part-assembly" relationship into two or three entities: one or two instances of `product_definition` and a corresponding `component_definition`. If there is only one instance of `product_definition`, then there is no separate 'design' for the part. If there are two instances of `product_definition` then one of them is the design and the other is the 'design usage'. This separation is controlled by a rule in `component_definition` that the `product_definition_formation` for `component_definition` and the `product_definition` that corresponds to the "part" concept is the same `product_definition_formation`. The correspondence is established by an instance of `product_definition_relationship` with a name of "instantiated part". The correspondence between the design and the design usage is established by an instance of `product_definition_relationship` with a name of 'design usage'.

5.2.1.8 Change management

This schema provides `action`, `action_directive`, `add_design_object_assignment`, `change`, `change_request`, `change_from_design_object_assignment`, `change_to_design_object_assignment`, `delete_object_assignment`, and `product_definition_relationship` entities to support design management processes. A data set that incorporates add, change, or delete shall include any required `action_assignment` subtype and the `product_definition_relationship` that identifies the previous `product_definition` and the current `product_definition`. Requests for change and specific change directions are provided.

5.2.1.9 Functional network definition

This schema provides `functional_unit`, `functional_unit_terminal_definition`, `component_functional_unit`, `component_functional_terminal`, and `network_node_definition` to exchange network listings (using a nodal formulation) of functional objects.

5.2.1.10 Geometric dimensioning and tolerancing

Each instance of `physical_unit_geometric_tolerance` along with supporting entity instances (e.g., `property_definition` with a description of "datum system property with material conditions") is intended to convey the data associated with a geometric tolerance requirement that is included in a single segment of a feature control frame of a technical drawing. Typically, feature control frames only have one segment. However, with composite geometric tolerances (i.e., composite position tolerance, composite linear profile tolerance, and composite surface profile tolerance) the feature control frame consists of an ordered set of segments. Associated with this set are certain order-dependent requirements as to the values that may be placed in the various compartments of the segments of these composite feature control frames.

EXAMPLE 1 If one or more data are specified in a lower segment of a composite feature control frame they must also be specified in segments that are listed above it.

To convey the data associated with a composite feature control frame, instances of `geometric_tolerance_relationship` are used to create an ordered set of `physical_unit_geometric_tolerance`. Additionally, constraints defined within the relevant `physical_unit_geometric_tolerance` (i.e., `position_tolerance`, `linear_profile_tolerance`, and `surface_profile_tolerance`) are used to specify the order-dependent requirements and to insure a valid data structure.

5.2.1.11 Physical network definition

This schema provides `physical_unit_network_definition`, `physical_connectivity_definition`, and `component_terminal` to exchange network listings (using a nodal formulation) of physical objects.

5.2.1.12 Product classification

This schema provides predefined product classifications of assembly, bare die, functionality, interconnect, material, package, physical network design, piece part, printed part, requirements model, simulation model, standard part, technology specific model, and template model. These predefined product classifications may require specific types of `product_definition` to support them or specific context values.

EXAMPLE 1 It is valid to have a package classification and to have an instance of `interconnect_definition` associated with it, but an instance of `package` will also exist that references that product instance.

5.2.1.13 Requirements definition

This schema provides `group_product_definition`, `group_shape_aspect`, `grouped_requirements_property`, `mounting_restriction_area`, `requirement_allocation_group`, `requirements_property`, and `requirements_property_group` to convey the data associated with technical requirements and the allocation of those requirements to product data.

5.2.2 electronic_assembly_interconnect_and_packaging_design type definitions

5.2.2.1 action_assigned_item

An action_assigned_item type specifies those items to which an action may be assigned.

EXPRESS specification:

```
*)
TYPE action_assigned_item = SELECT
    (product_definition_formation);
END_TYPE;
(*
```

5.2.2.2 approval_assigned_item

An approval_assigned_item type specifies those items to which an approval may be assigned which has more values than approved, unapproved.

EXPRESS specification:

```
*)
TYPE approval_assigned_item = SELECT
    (action,
     action_directive,
     certification,
     change,
     change_request,
     configuration_effectivity,
     configuration_item,
     contract,
     date,
     directed_action,
     document,
     product,
     product_definition,
     product_definition_formation,
     product_definition_formation_with_specified_source,
     representation,
     security_classification,
     start_request,
     start_work,
     versioned_action_request);
END_TYPE;
(*
```

5.2.2.3 certification_assigned_item

A certification_assigned_item type specifies those items to which a certification may be assigned.

EXPRESS specification:

```
*)
TYPE certification_assigned_item = SELECT
    (supplied_part_relationship,
     make_from_usage_option,
     product_definition_formation_with_specified_source);
END_TYPE;
(*
```

5.2.2.4 change_request_item

A change_request_item type specifies those items to which a request for change may be assigned.

EXPRESS specification:

```
*)
TYPE change_request_item = SELECT
    (product_definition_formation,
     representation_relationship);
END_TYPE;
(*
```

5.2.2.5 classification_assigned_item

A classification_assigned_item type specifies those items to which a classification may be assigned.

EXPRESS specification:

```
*)
TYPE classification_assigned_item = SELECT
    (material_designation,
     requirements_property,
     signal,
     stratum);
END_TYPE;
(*
```


5.2.2.6 contract_assigned_item

A contract_assigned_item type specifies those objects to which a contract may be assigned.

EXPRESS specification:

```
*)
TYPE contract_assigned_item = SELECT
    (alternate_product_relationship,
     directed_action,
     product,
     product_definition_formation);
END_TYPE;
(*
```

5.2.2.7 date_and_time_assigned_item

A date_and_time_assigned_item type specifies those items to which a date and a time may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_time_assigned_item = SELECT
    (approval_person_organization,
     certification,
     change,
     contract,
     directed_action,
     document,
     product_definition,
     rule_action,
     security_classification,
     versioned_action_request);
END_TYPE; -- date_and_time_assigned_item
(*
```

5.2.2.8 date_assigned_item

A date_assigned_item type specifies those items to which a date may be assigned.

EXPRESS specification:

```
*)  
TYPE date_assigned_item = SELECT  
    (product_definition,  
     versioned_action_request,  
     directed_action,  
     approval_person_organization,  
     contract,  
     security_classification,  
     certification);  
END_TYPE; -- date_assigned_item  
(*
```

5.2.2.9 document_assigned_item

A document_assigned_item type specifies those items to which a document may be assigned.

EXPRESS specification:

```

*)
TYPE document_assigned_item = SELECT
    (action_method,
     assembly_component_usage,
     bond_category,
     characterized_object,
     component_definition,
     component_feature,
     component_interface_terminal,
     component_shape_aspect,
     component_terminal,
     coordinated_representation_item,
     configuration_item,
     descriptive_representation_item,
     externally_defined_item,
     functional_unit_terminal_definition,
     inter_stratum_feature,
     interconnect_module_component_surface_feature,
     layer,
     laminate_component_interface_terminal,
     material_designation,
     measure_representation_item,
     package_terminal,
     part_template_definition,
     physical_connectivity_definition,
     physical_connectivity_element,
     product,
     product_definition,
     product_definition_formation,
     product_definition_formation_relationship,
     property_definition,
     representation,
     representation_item,
     requirements_property,
     rule_definition,
     shape_aspect,
     signal,
     stratum,
     stratum_feature);
END_TYPE;
( *

```

5.2.2.10 document_identifier_assigned_item

A document_identifier_assigned_item type specifies those items to which a document_identifier may be assigned.

EXPRESS specification:

```
* )  
TYPE document_identifier_assigned_item = SELECT  
    (document);  
END_TYPE;  
(*
```

5.2.2.11 frozen_assigned_item

An frozen_assigned_item type specifies those items to which an approval with value of (approved, unapproved) may be assigned to indicate acceptance of a design element.

EXPRESS specification:

```

*)
TYPE frozen_assigned_item = SELECT
  (alternate_product_relationship,
   applied_date_and_time_assignment,
   applied_document_reference,
   applied_group_assignment,
   characterized_object,
   colour,
   component_definition,
   component_shape_aspect,
   composite_curve_segment,
   curve_style,
   curve_style_font,
   curve_style_font_and_scaling,
   dimension_related_tolerance_zone_element,
   dimensional_characteristic_representation,
   dimensional_size,
   document_usage_constraint,
   externally_defined_item,
   fill_area_style,
   fill_area_style_colour,
   geometric_tolerance,
   geometric_tolerance_relationship,
   group,
   inter_stratum_feature,
   material_designation,
   measure_with_unit,
   plus_minus_tolerance,
   pre_defined_item,
   product_concept,
   product_definition_formation_relationship,
   product_definition_relationship,
   property_definition,
   property_definition_relationship,
   property_definition_representation,
   representation,
   representation_context,
   representation_item,
   representation_relationship,
   runout_zone_orientation,
   shape_aspect,
   shape_aspect_relationship,
   stratum_feature,
   tolerance_value,
   tolerance_zone_form);
END_TYPE;
( *

```

5.2.2.12 group_assigned_item

A group_assigned_item type specifies those items to which a group may be assigned.

EXPRESS specification:

```

*)
TYPE group_assigned_item = SELECT
    (assembly_bond_definition,
     design_object,
     component_definition,
     component_feature,
     component_interface_terminal,
     component_shape_aspect,
     component_terminal,
     compound_representation_item,
     configuration_item,
     inter_stratum_feature,
     laminate_component_interface_terminal,
     printed_part_template_terminal,
     product,
     product_definition_formation,
     property_definition_representation,
     representation,
     representation_item,
     requirements_property,
     rule_definition,
     shape_aspect,
     shape_aspect_relationship,
     stratum,
     stratum_feature,
     stratum_surface);
END_TYPE;
( *
```

5.2.2.13 managed_design_object

A managed_design_object type specifies those items to which design management information may be assigned.

EXPRESS specification:

```

*)
TYPE managed_design_object = SELECT
  (action,
   alternate_product_relationship,
   applied_approval_assignment,
   applied_date_and_time_assignment,
   applied_document_reference,
   applied_group_assignment,
   approval,
   approval_status,
   certification,
   characterized_object,
   colour,
   composite_curve_segment,
   configuration_effectivity,
   configuration_item,
   contract,
   curve_style,
   curve_style_font,
   curve_style_font_and_scaling,
   date,
   dimension_related_tolerance_zone_element,
   dimensional_characteristic_representation,
   dimensional_size,
   document,
   document_usage_constraint,
   externally_defined_item,
   fill_area_style,
   fill_area_style_colour,
   geometric_tolerance,
   geometric_tolerance_relationship,
   group,
   layer_connection_point,
   layer_connection_point_link,
   material_designation,
   measure_with_unit,
   organization,
   person,
   person_and_organization,
   plus_minus_tolerance,
   pre_defined_item,
   product,
   product_concept,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_relationship,
   property_definition,
   property_definition_relationship,

```

```

        property_definition_representation,
        representation,
        representation_context,
        representation_item,
        representation_relationship,
        runout_zone_orientation,
        security_classification,
        shape_aspect,
        shape_aspect_relationship,
        tolerance_value,
        tolerance_zone_form,
        versioned_action_request);
END_TYPE; -- managed_design_object
(*)

```

5.2.2.14 organization_assigned_item

An organization_assigned_item type specifies those items to which an organization may be assigned.

EXPRESS specification:

```

*)
TYPE organization_assigned_item = SELECT
    (configuration_item,
    contract,
    directed_action,
    document,
    versioned_action_request,
    product,
    product_definition,
    product_definition_formation,
    product_definition_formation_with_specified_source,
    rule_definition,
    representation,
    security_classification);
END_TYPE; -- organization_assigned_item
(*)

```

5.2.2.15 owner_assigned_item

An owner_assigned_item type specifies those items to which a person_organization may be assigned to indicate direct control of a design element.

NOTE Owner in this usage indicates a single point of design control.

EXPRESS specification:

```

*)
TYPE owner_assigned_item = SELECT
  (alternate_product_relationship,
   characterized_object,
   colour,
   component_definition,
   component_shape_aspect,
   composite_curve_segment,
   curve_style,
   curve_style_font,
   curve_style_font_and_scaling,
   dimension_related_tolerance_zone_element,
   dimensional_characteristic_representation,
   dimensional_size,
   document_usage_constraint,
   externally_defined_item,
   fill_area_style,
   fill_area_style_colour,
   geometric_tolerance,
   geometric_tolerance_relationship,
   group,
   inter_stratum_feature,
   material_designation,
   measure_with_unit,
   plus_minus_tolerance,
   pre_defined_item,
   product_concept,
   product_definition_formation_relationship,
   product_definition_relationship,
   property_definition,
   property_definition_relationship,
   property_definition_representation,
   representation,
   representation_context,
   representation_item,
   representation_relationship,
   runout_zone_orientation,
   shape_aspect,
   shape_aspect_relationship,
   stratum_feature,
   tolerance_value,
   tolerance_zone_form);
END_TYPE; -- owner_assigned_item
( *

```

5.2.2.16 person_and_organization_assigned_item

A `person_and_organization_assigned_item` type specifies those items to which a person associated with an organization may be assigned.

EXPRESS specification:

```
*)
TYPE person_and_organization_assigned_item = SELECT
    (change,
     configuration_item,
     contract,
     document,
     product,
     product_definition_formation,
     product_definition,
     directed_action,
     rule_action,
     security_classification,
     versioned_action_request);
END_TYPE; -- person_and_organization_assigned_item
(*
```

5.2.2.17 person_assigned_item

A person_assigned_item type specifies those items to which a person may be assigned.

EXPRESS specification:

```
*)
TYPE person_assigned_item = SELECT
    (document);
END_TYPE;
(*
```

5.2.2.18 promissory_usage_in_product_model_assigned_item

A promissory_usage_in_product_model_assigned_item type specifies those items to which a promissory_usage_in_product_model_assignment may be made.

EXPRESS specification:

```
*)
TYPE promissory_usage_in_product_model_assigned_item = SELECT
    (product_concept,
     product_definition);
END_TYPE;
(*
```

5.2.2.19 replacement_from_item

A replacement_from_item type specifies those items for which a replacement may be specified.

EXPRESS specification:

```
*)
TYPE replacement_from_item = SELECT
    (product_definition_formation);
END_TYPE;
(*
```

5.2.2.20 replacement_to_item

A replacement_to_item type specifies those items which may be used as a replacement for another item.

EXPRESS specification:

```
*)
TYPE replacement_to_item = SELECT
    (product_definition_formation);
END_TYPE;
(*
```

5.2.2.21 security_classification_assigned_item

A security_classification_assigned_item type specifies those items to which a security classification may be assigned.

EXPRESS specification:

```
*)
TYPE security_classification_assigned_item = SELECT
    (applied_promissory_usage_in_product_model_assignment,
     assembly_component_usage,
     document,
     make_from_usage_option,
     product_definition_formation,
     product_definition);
END_TYPE;
(*
```

5.2.2.22 start_request_item

A start_request_item type specifies those items to which a request to start may be assigned.

EXPRESS specification:

```
*)
TYPE start_request_item = SELECT
    (product_definition_formation,
     representation_relationship);
END_TYPE;
( *
```

5.2.2.23 work_item

A work_item type specifies those items to which work may be performed.

EXPRESS specification:

```
*)
TYPE work_item = SELECT
    (product_definition_formation);
END_TYPE;
( *
```

5.2.3 electronic_assembly_interconnect_and_packaging_design entities

5.2.3.1 electronic_assembly_interconnect_and_packaging_design entity definitions

5.2.3.1.1 absorbed_dose_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An absorbed_dose_measure_with_unit is a measure_with_unit that implements the ARM concept of Absorbed_dose_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY absorbed_dose_measure_with_unit
    SUBTYPE OF (measure_with_unit);
WHERE
```

```

WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\absorbed_dose_measure_with_unit))) = 0;

WR2: SELF\measure_with_unit.unit_component\si_unit.name = gray;

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- absorbed_dose_measure_with_unit
(*)

```

Formal propositions:

WR1: The absorbed_dose_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The absorbed_dose_measure_with_unit shall reference an si_unit with the name of gray.

WR3: The unit_component shall be an si_unit.

5.2.3.1.2 activity_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An activity_measure_with_unit is a measure_with_unit that implements the ARM concept of Activity_-measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY activity_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\activity_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = becquerel;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- activity_measure_with_unit
(*)

```

Formal propositions:

WR1: The activity_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The activity_measure_with_unit shall reference a becquerel si_unit.

WR3: The unit_component shall be an si_unit.

5.2.3.1.3 add_design_object_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An add_design_object_assignment is an action_assignment that implements the ARM concept of Add_design_object_management_relationship (see 4.2.3).

EXPRESS specification:

```
*)
ENTITY add_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(it))
    AND (it\product_definition_relationship.name =
      'design object addition')))) = 1;
END_ENTITY; -- add_design_object_assignment
(*
```

Formal propositions:

WR1: The add_design_object_assignment shall have exactly one item that is a product_definition_relationship with a name of 'design object addition'.

Informal propositions:

IP1: The add_design_object_assignment shall have exactly one item that is a managed_design_object that is not a product_definition_relationship with a name of 'design object addition' and is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design.

5.2.3.1.4 add_design_object_request_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `add_design_object_request_assignment` is an `action_request_assignment` that implements the ARM concepts of `Add_design_object_management_relationship` (see 4.2.3) and `Change_request` (see 4.2.97).

EXPRESS specification:

```

*)
ENTITY add_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(it))
    AND (it\product_definition_relationship.name =
      'design object addition')) = 1;
END_ENTITY; -- add_design_object_request_assignment
(

```

Formal propositions:

WR1: The `add_design_object_request_assignment` shall have exactly one item that is a `product_definition_relationship` with a name of 'design object addition'.

Informal propositions:

IP1: The `add_design_object_request_assignment` shall have exactly one item that is a `managed_design_object` that is not a `product_definition_relationship` with a name of 'design object addition' and is not referenced by another `product_definition_relationship` with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design.

5.2.3.1.5 **altered_package_terminal**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `altered_package_terminal` is a `package_terminal` that implements the ARM concept of `Altered_package_terminal` (see 4.2.9).

EXPRESS specification:

```

*)
ENTITY altered_package_terminal
  ABSTRACT SUPERTYPE OF ((shape_formed_terminal ANDOR
    surface_prepped_terminal) ANDOR length_trimmed_terminal)
  SUBTYPE OF (package_terminal);
WHERE

  WR1: SIZEOF(QUERY ( sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'terminal to be altered' ))
    = 1;

  WR2: (SELF.of_shape.definition\product_definition.description =
    'altered package');
END_ENTITY; -- altered_package_terminal
(

```

Formal propositions:

WR1: The `altered_package_terminal` shall be referenced by one `shape_aspect_relationship` with a description of 'terminal to be altered' as the `related_shape_aspect`.

WR2: The `altered_package_terminal` shall be an aspect of the shape of a `product_definition` with a description of 'altered package'.

5.2.3.1.6 analytical_model

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `analytical_model` is a representation that implements the ARM concept of `Analytical_model` (see 4.2.14).

EXPRESS specification:

```

*)
ENTITY analytical_model
  SUBTYPE OF (representation);
UNIQUE
  UR1: SELF\representation.name;
WHERE

```



```

WR1: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 2;

WR2: sizeof (QUERY (dr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    dr.assigned_document.kind\document_type.product_data_type =
    'language reference manual')) = 1;

WR3: sizeof (QUERY (it <* SELF.items |
    NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODEL_PARAMETER' IN
    TYPEOF (it)))) = 0;

WR4: sizeof (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') |
    (rr\representation_relationship.name = 'access mechanism') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANALYTICAL_MODEL_PORT' IN
    TYPEOF (rr.rep_2)))) >= 1;
END_ENTITY; -- analytical_model
( *

```

Formal propositions:

UR1: The name shall be unique within a population of analytical_model.

WR1: The analytical_model shall be referenced by exactly two document_references.

WR2: The analytical_model shall be referenced by exactly one document_reference where the document_type is 'language reference manual'.

WR3: The analytical_model shall have at least one model_parameter in its set of items.

WR4: The analytical_model shall be referenced as rep_1 in at least one representation_relationship with a name of 'access mechanism' where the rep_2 references an analytical_model_port.

5.2.3.1.7 analytical_model_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An analytical_model_definition is a product_definition that implements the ARM concepts of: Analytical_model_definition (see 4.2.15)

EXPRESS specification:

```

*)
ENTITY analytical_model_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF (SELF))) OR
    (SIZEOF (QUERY (docs <*
    SELF\product_definition_with_associated_documents.
    documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);

  WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) +
    SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ada.role\date_role.name = 'creation date')) = 1;

  WR3: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

  WR4: SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

  WR5: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

  WR6: SIZEOF (TYPEOF (SELF) - (TYPEOF (SELF\product_definition ||
    SELF\analytical_model_definition))) = 0;

```

```

WR7: SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
    'PRODUCTS')) |
    prpc\product_category.name = 'simulation model')) >= 1;
END_ENTITY; -- analytical_model_definition
(*)

```

Formal propositions:

WR1: If the analytical_model_definition is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR2: The analytical_model_definition shall have exactly one creation date.

WR3: The analytical_model_definition shall have exactly one approval.

WR4: The analytical_model_definition shall have at least one organization or person_and_organization as the creator.

WR5: The analytical_model_definition shall have exactly one security_classification.

WR6: The analytical_model_definition shall not be any other type of product_definition.

WR7: The analytical_model_definition shall have a product_category name of 'simulation model'.

5.2.3.1.8 analytical_model_make_from_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An analytical_model_make_from_relationship is a representation and a representation_relationship that implements the ARM concept of Analytical_model_make_from (see ??).

EXPRESS specification:

```

*)
ENTITY analytical_model_make_from_relationship
    SUBTYPE OF (representation, representation_relationship);
UNIQUE
    UR1: SELF\representation_relationship.rep_1, SELF\representation_relationship.rep_2;
WHERE

    WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL' ] *
        TYPEOF (SELF.rep_1)) = 1;

```

```

WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'ANALYTICAL_MODEL' ] *
  TYPEOF (SELF.rep_2)) = 1;

WR3: SIZEOF( QUERY(rr <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REPRESENTATION_RELATIONSHIP.REP_1') |NOT(
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MAKE_FROM_MODEL_PORT_RELATIONSHIP' IN TYPEOF(rr.rep_2)))))) = 0;

WR4: SIZEOF( QUERY(rr <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REPRESENTATION_RELATIONSHIP.REP_1') |NOT(
  (rr\representation_relationship.name = 'associated make from') AND
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MAKE_FROM_MODEL_PORT_RELATIONSHIP' IN TYPEOF(rr.rep_2)))))) = 0;

WR5: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'ANALYTICAL_MODEL' ] *
  TYPEOF (SELF)) = 0;

WR6: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
  SELF\representation_relationship ||
  SELF\analytical_model_make_from_relationship))) = 0;
END_ENTITY; -- analytical_model_make_from_relationship
(*

```

Formal propositions:

UR1: The combination of rep_1 and rep_2 shall be unique within a population of analytical_model_-make_from_relationship.

WR1: The analytical_model_make_from_relationship shall reference an analytical_model as its rep_1.

WR2: The analytical_model_make_from_relationship shall reference an analytical_model as its rep_2.

WR3: The analytical_model_make_from_relationship shall be related to only a make_from_model_-port_relationship via a representation_relationship.

WR4: The analytical_model_make_from_relationship shall be related to only a make_from_model_-port_relationship via a representation_relationship with a name of 'associated make from'.

WR5: The analytical_model_make_from_relationship shall not be an analytical_model.

WR6: The analytical_model_make_from_relationship shall only be a representation, representation_item or in addition to itself.

5.2.3.1.9 analytical_model_port

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An analytical_model_port is a representation that describes an interface point for an analytical_model.
An analytical_model_port implements the ARM constructs of:

- Analytical_model_port (see 4.2.17);
- Analytical_model_scalar_port (see 4.2.20);
- Analytical_model_vector_port (see 4.2.21);
- Digital_analytical_model_port (see 4.2.290).

EXPRESS specification:

```

*)
ENTITY analytical_model_port
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'distributed property') AND
    (it\descriptive_representation_item.description IN
    ['true', 'false']))) = 1;

  WR2: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'nominal signal flow direction') AND
    (it\descriptive_representation_item.description IN
    ['input direction',
    'output direction', 'bidirectional', 'unknown direction',
    'not applicable']))) <= 1;

  WR3: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'port type') AND
    (it\descriptive_representation_item.description IN
    ['string property type',
    'logical property type', 'physical property type',
    'boolean property type']))) = 1;

```

```

WR4: sizeof (QUERY (am <* QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    rr\representation_relationship.name = 'access mechanism') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANALYTICAL_MODEL' IN
    TYPEOF (am.rep_1))) = 1;

WR5: sizeof (QUERY (aga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    aga.assigned_group\group.name IN ['scalar port', 'digital scalar port',
    'vector port', 'digital vector port',
    'digital analytical model port'])) <= 1;

WR6: (NOT (sizeof (QUERY (aga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    aga.assigned_group\group.name IN ['vector port', 'digital vector port']))
    = 1)) OR
    (sizeof (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'size') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COUNT_MEASURE' IN
    TYPEOF (it\measure_with_unit.value_component)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONTEXT_DEPENDENT_UNIT' IN
    TYPEOF (it\measure_with_unit.unit_component)))) = 1);

WR7: (NOT (sizeof (QUERY (aga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    aga.assigned_group\group.name IN ['digital scalar port',
    'digital vector port', 'digital analytical model port'])) = 1)) OR
    (sizeof (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'nominal signal flow direction') AND
    (it\descriptive_representation_item.description IN ['input direction',
    'output direction', 'bidirectional', 'unknown direction']))) = 1);
END_ENTITY; -- analytical_model_port
(

```

Formal propositions:

WR1: The analytical_model_port shall have exactly one descriptive_representation_item with a name of 'distributed property' and a description of 'true' or 'false'.

WR2: The analytical_model_port shall have at most one descriptive_representation_item with a name of 'nominal signal flow direction' and a description of 'input direction', 'output direction', 'bidirectional', 'unknown direction', or 'not applicable'.

WR3: The analytical_model_port shall have exactly one descriptive_representation_item with a name of 'port type' and a description of 'string property type', 'logical property type', 'physical property type', or 'boolean property type'.

WR4: The analytical_model_port shall be related to exactly one analytical_model as the 'access mechanism'.

WR5: The analytical_model_port shall be assigned to at most one group with a name of 'scalar port', 'digital scalar port', 'vector port', 'digital vector port', or 'digital analytical model port'.

WR6: If the analytical_model_port is assigned to a group with a name of 'vector port' or 'digital vector port', it shall have exactly one measure_representation_item with a name of 'size', a value_component that is a count_measure, and a unit_component that is a context_dependent_unit.

WR7: If the analytical_model_port is assigned to a group with a name of 'digital scalar port', 'digital vector port', or 'digital analytical model port', it shall have exactly one descriptive_representation_item with a name of 'nominal signal flow direction' and a description of 'input direction', 'output direction', 'bidirectional', or 'unknown direction'.

5.2.3.1.10 analytical_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An analytical_representation is a representation that associates specific properties with an analytical_model. An analytical_representation implements the ARM concept of Analytical_representation (see 4.2.22).

EXPRESS specification:

```
*)
ENTITY analytical_representation
  SUBTYPE OF (representation);
  UNIQUE
    UR1: SELF\representation.name;
  WHERE
```

```

WR1: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PARAMETER_ASSIGNMENT_REPRESENTATION' IN
    TYPEOF (rr.rep_2))) >= 1;

WR2: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANALYTICAL_MODEL' IN TYPEOF (rr.rep_1))) = 1;
END_ENTITY; -- analytical_representation
(*

```

Formal propositions:

UR1: The name shall be unique within a population of analytical_representation.

WR1: The analytical_representation shall have at least one representation_relationship that references a parameter_assignment_representation.

WR2: The analytical_representation shall have exactly one representation_relationship that references an analytical_model.

5.2.3.1.11 angular_dimension_with_orientation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An angular_dimension_with_orientation is a dimensional_location that defines the angle between two shape elements. An angular_dimension_with_orientation implements the ARM concept of Angular_dimension_with_direction_vector (see 4.2.24).

EXPRESS specification:

```

*)
ENTITY angular_dimension_with_orientation
    SUBTYPE OF (dimensional_location);
WHERE

    WR1: SELF\shape_aspect_relationship.description = 'angular';
END_ENTITY; -- angular_dimension_with_orientation
(*

```


Formal propositions:

WR1: The description of the angular_dimension_with_orientation shall be 'angular'.

Informal propositions:

IP1: The angular_dimension_with_orientation shall have exactly one property_definition with a description of 'dimensional location property', and that property_definition either shall be represented by a representation that contains a placement with a name of 'orientation' or shall relate to a property_definition with a description of 'datum based vector orientation' to specify the 'measurement orientation'.

IP2: If the angular_dimension_with_orientation has a property_definition with a description of 'dimensional location property' and that property_definition relates to a property_definition with a description of 'datum based vector orientation' to specify the 'measurement orientation', the combination of the shape_aspect specified as the related_shape_aspect, the shape_aspect specified as the relating_shape_aspect, and the property_definition with a description of 'datum based vector orientation' shall be unique.

IP3: If the angular_dimension_with_orientation has a property_definition with a description of 'dimensional location property' and that property_definition is represented by a representation that contains a placement with a name of 'orientation', the combination of the shape_aspect specified as the related_shape_aspect, the shape_aspect specified as the relating_shape_aspect, and the placement with a name of 'orientation' shall be unique.

5.2.3.1.12 angularity_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An angularity_tolerance is a geometric_tolerance_with_specified_datum_system. An angularity_tolerance implements the ARM concept of Angularity_tolerance (see 4.2.26).

EXPRESS specification:

```
*)
ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'angularity';
END_ENTITY; -- angularity_tolerance
(*)
```

Formal propositions:

WR1: The name for the angularity_tolerance shall be 'angularity'.

Informal propositions:

IP1: The `toleranced_shape_aspect` of the `angularity_tolerance` shall either lie on the boundary of the shape of the product; be a `centre_of_symmetry` with a description of either 'axis' or 'plane'; be a tangent with a description of 'plane'; be a `part_template_definition`; or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which either are all of type `part_template_definition`, are all of type `centre_of_symmetry` and have the same description – either 'axis' or 'plane', or are all of type tangent with a description of 'plane'.

IP2: The `angularity_tolerance` shall have at most two `physical_unit_datum` in its referenced `datum_system`.

IP3: The `angularity_tolerance` shall have at least one `physical_unit_datum` in its referenced `datum_system` that has description of either 'axis' or 'plane'.

IP4: If the `toleranced_shape_aspect` of the `angularity_tolerance` is neither a `centre_of_symmetry` with a description of 'axis' nor a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'axis', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

5.2.3.1.13 **applied_action_assignment**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_action_assignment` is an `action_assignment` that specifies a set of `action_assigned_item`.

EXPRESS specification:

```
* )
ENTITY applied_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_assigned_item;
END_ENTITY; -- applied_action_assignment
(*
```

5.2.3.1.14 **applied_approval_assignment**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_approval_assignment` is an `approval_assignment` that specifies a set of `approval_assigned_item`.

EXPRESS specification:

```

*)
ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approval_assigned_item;
END_ENTITY; -- applied_approval_assignment
( *

```

5.2.3.1.15 applied_certification_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An applied_certification_assignment is a certification_assignment that specifies a set of certification_assigned_item.

EXPRESS specification:

```

*)
ENTITY applied_certification_assignment
  SUBTYPE OF (certification_assignment);
  items : SET [1:?] OF certification_assigned_item;
END_ENTITY; -- applied_certification_assignment
( *

```

5.2.3.1.16 applied_classification_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An applied_classification_assignment is a group_assignment that specifies a set of classification_assigned_item.

EXPRESS specification:

```

*)
ENTITY applied_classification_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF classification_assigned_item;
END_ENTITY; -- applied_classification_assignment
( *

```

5.2.3.1.17 applied_contract_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_contract_assignment` is a `contract_assignment` that specifies a set of `contract_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_contract_assignment
  SUBTYPE OF (contract_assignment);
  items : SET [1:?] OF contract_assigned_item;
END_ENTITY; -- applied_contract_assignment
(*
```

5.2.3.1.18 `applied_date_and_time_assignment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_date_and_time_assignment` is a `date_and_time_assignment` that specifies a set of `date_and_time_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET [1:?] OF date_and_time_assigned_item;
WHERE

  WR1: applied_date_time_correlation
    (SELF, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN');

  WR2: (NOT (SELF.role\date_time_role.name = 'participant date and time')) OR
    (SIZEOF (QUERY (ra <* QUERY (it <* SELF.items |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_ACTION' IN
      TYPEOF (it)) |
      NOT (SIZEOF (USEDIN (ra,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')) = 1))) = 0));
END_ENTITY; -- applied_date_and_time_assignment
(*
```

Formal propositions:

WR1: The role of the `applied_date_and_time_assignment` shall correlate with the items of the `applied_date_and_time_assignment`.

WR2: If the `applied_date_and_time_assignment` has a role of 'participant date and time', each `rule_action` in the set of items shall have exactly one `applied_person_and_organization_assignment`.

5.2.3.1.19 `applied_date_assignment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_date_assignment` is a `date_assignment` that specifies a set of `date_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_assigned_item;
WHERE

  WR1: applied_date_correlation (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN');
END_ENTITY; -- applied_date_assignment
(*
```

Formal propositions:

WR1: The role of the `applied_date_assignment` shall correlate with the items of the `applied_date_assignment`.

5.2.3.1.20 `applied_document_reference`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_document_reference` is a `document_reference` that specifies a set of `document_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_assigned_item;
END_ENTITY; -- applied_document_reference
(*
```

5.2.3.1.21 `applied_group_assignment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_group_assignment` is a `group_assignment` that specifies a set of `group_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_group_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF group_assigned_item;
END_ENTITY; -- applied_group_assignment
(*
```

Informal propositions:

IP1: If the `applied_group_assignment` has an `assigned_group` that is a `functional_terminal_group` then it shall only have items that are `functional_unit_terminal_definitions` with a description of 'scalar terminal'.

5.2.3.1.22 `applied_organization_assignment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_organization_assignment` is an `organization_assignment` that specifies a set of `organization_assigned_item`.

EXPRESS specification:

```
*)
ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_assigned_item;
END_ENTITY; -- applied_organization_assignment
(*
```

5.2.3.1.23 `applied_person_and_organization_assignment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_person_and_organization_assignment` is a `person_and_organization_assignment` that specifies a set of `person_and_organization_assigned_item`.

EXPRESS specification:

```

*)
ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_assigned_item;
END_ENTITY; -- applied_person_and_organization_assignment
( *

```

5.2.3.1.24 applied_person_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An applied_person_assignment is a person_assignment that specifies a set of person_assigned_item.

EXPRESS specification:

```

*)
ENTITY applied_person_assignment
  SUBTYPE OF (person_assignment);
  items : SET [1:?] OF person_assigned_item;
END_ENTITY; -- applied_person_assignment
( *

```

5.2.3.1.25 applied_promissory_usage_in_product_model_assignment

An applied_promissory_usage_in_product_model_assignment is a group_assignment that specifies a set of promissory_usage_in_product_model_assigned_item. One item shall be a product_concept and the other item shall be a product_definition.

EXPRESS specification:

```

*)
ENTITY applied_promissory_usage_in_product_model_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF promissory_usage_in_product_model_assigned_item;
WHERE

  WR1: SIZEOF(QUERY(pm <* items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_CONCEPT' IN TYPEOF(pm)
  )) = 1;

```

```

WR2: sizeof(QUERY(pd <* items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION' IN TYPEOF(pd)
    )) = 1;

WR3: sizeof(items) = 2;

WR4: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROMISSORY_USAGE_IN_PRODUCT_MODEL' IN TYPEOF(SELF\group_assignment.assigned_group)

WR5: sizeof(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) <= 1;
END_ENTITY; -- applied_promissory_usage_in_product_model_assignment
(*)

```

5.2.3.1.26 applied_security_classification_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `applied_security_classification_assignment` is a `security_classification_assignment` that specifies a set of `security_classification_assigned_item`.

EXPRESS specification:

```

*)
ENTITY applied_security_classification_assignment
    SUBTYPE OF (security_classification_assignment);
    items : SET [1:?] OF security_classification_assigned_item;
END_ENTITY; -- applied_security_classification_assignment
(*)

```

5.2.3.1.27 array_placement_group_component_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `array_placement_group_component_definition` is a `assembly_group_component_definition` that implements the ARM concept of `Array_placement_group` (see 4.2.29) .

EXPRESS specification:

```

*)
ENTITY array_placement_group_component_definition
    SUPERTYPE OF (ONEOF(linear_array_placement_group_component_definition,
                        rectangular_array_placement_group_component_definition))
    SUBTYPE OF (assembly_group_component_definition);
END_ENTITY; -- array_placement_group_component_definition
( *

```

5.2.3.1.28 array_placement_group_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An array_placement_group_component_shape_aspect is a assembly_group_component_shape_aspect that implements the ARM concepts of Array_placement_group (see 4.2.29) .

EXPRESS specification:

```

*)
ENTITY array_placement_group_component_shape_aspect
    SUPERTYPE OF (ONEOF(linear_array_placement_group_component_shape_aspect,
                        rectangular_array_placement_group_component_shape_aspect))
    SUBTYPE OF (assembly_group_component_shape_aspect);
END_ENTITY; -- array_placement_group_component_shape_aspect
( *

```

5.2.3.1.29 assembly_bond_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An assembly_bond_definition is a shape_aspect that defines a joint that uses material to bond together portions of the assembly. An assembly_bond_definition implements the ARM concept of Assembly_bond_definition (see 4.2.33).

EXPRESS specification:

```

*)
ENTITY assembly_bond_definition
    SUBTYPE OF (shape_aspect);
UNIQUE
    UR1: SELF\shape_aspect.name;
WHERE

```

```

WR1: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_DESIGNATION.DEFINITIONS')) = 1;

WR2: sizeof (QUERY (aga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOND_CATEGORY' IN
    TYPEOF (aga.assigned_group))) = 1;

WR3: sizeof (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'bonded feature 1' )) = 1;

WR4: sizeof (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'bonded feature 2' )) = 1;

WR5: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PRODUCT_DEFINITION' IN TYPEOF (SELF.of_shape.definition))
    AND
    (sizeof (QUERY (prpc <* USEDIN (SELF.of_shape.definition.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
    'PRODUCTS') |
    prpc\product_category.name = 'technology specific model')) >= 1));
END_ENTITY; -- assembly_bond_definition
(*

```

Formal propositions:

UR1: The name of the assembly_bond_definition shall be unique.

WR1: The assembly_bond_definition shall have exactly one material_designation.

WR2: The assembly_bond_definition shall be in exactly one bond_category.

WR3: The assembly_bond_definition shall be related to exactly one shape_aspect as the 'bonded feature 1'.

WR4: The assembly_bond_definition shall be related to exactly one shape_aspect as the 'bonded feature 2'.

WR5: The assembly_bond_definition shall be an aspect of the shape of the externally_defined_product_definition of a 'technology specific model' product.

5.2.3.1.30 assembly_definition

An assembly_definition is a physical_unit. An assembly_definition implements the ARM concepts of a Pca (see 4.2.639) and a Pca_usage_view (see 4.2.641).

EXPRESS specification:

```

*)
ENTITY assembly_definition
  SUBTYPE OF (physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: NOT EXISTS(SELF\product_definition.name) OR
        (SELF\product_definition.name = 'assembly module');

  WR3: (NOT (SELF.frame_of_reference.name = 'physical design')) OR
        (SIZEOF (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_DEFINITION_RELATIONSHIP.' +
          'RELATED_PRODUCT_DEFINITION') |
          pdr\product_definition_relationship.name = 'design usage') |
        (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'ASSEMBLY_DEFINITION',
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION',
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'LIBRARY_DEFINED_ASSEMBLY_DEFINITION'] *
        TYPEOF (du.relateing_product_definition)) = 1) AND
        (du.relateing_product_definition.frame_of_reference.name =
          'physical design usage') AND
        (du.relateing_product_definition\product_definition.name = 'assembly module')
        )) = 1);

  WR4: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
          NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
          ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'ASSEMBLY_MODULE_TERMINAL' IN
          TYPEOF (sa)) AND
          (sa\shape_aspect.description = 'pca terminal')))) >= 2))) = 0);
END_ENTITY; -- assembly_definition
(

```

Formal propositions:

WR1: The name shall exist.

WR2: The assembly_definition shall be a definition of a product that has a name of 'assembly module'.

WR3: If the assembly_definition is a physical design, it shall relate to exactly one assembly_definition, externally_defined_assembly_definition, or library_defined_assembly_definition that is a physical design usage with a name of 'assembly module' as the design usage.

WR4: If the assembly_definition has a frame_of_reference name of 'physical design usage', it shall have at least two shape_aspects that are assembly_module_terminals with a description of 'pca terminal'.

5.2.3.1.31 assembly_group_component_definition

An assembly_group_component_definition is a component_definition that implements the ARM concept of Assembly_group_component (see 4.2.40) .

EXPRESS specification:

```

*)
ENTITY assembly_group_component_definition
  SUBTYPE OF (component_definition);
WHERE

  WR1: (SIZEOF (QUERY (gc <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'group component') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (gc.related_product_definition)))) >= 1) OR
    (SIZEOF (QUERY(agcsa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_GROUP_COMPONENT_SHAPE_ASPECT' IN TYPEOF (agcsa)))) = 0);

```

```

WR2: (SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))) = 0) OR
(SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)) AND
(SIZEOF (QUERY( acu <* USEDIN (aj\shape_aspect_relationship.relatng_shape_aspect,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (acu))
AND (acu\product_definition_relationship.relatng_product_definition = SELF)
)) >= 1))) >= 1);

WR3: (SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))) = 0) OR
(SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj))AND
(SIZEOF (QUERY( acu <* USEDIN (aj\
shape_aspect_relationship.related_shape_aspect,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (acu))
AND (acu\product_definition_relationship.relatng_product_definition = SELF)
)) = 0))) = 0);

```

```

WR4: (SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))) = 0) OR
(SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)) AND
(SIZEOF (QUERY( cl <* USEDIN (aj\
shape_aspect_relationship.related_shape_aspect.of_shape.definition,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REPRESENTATION.ITEMS') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_LOCATION' IN TYPEOF (cl))
AND (aj IN cl\representation.items)))) = 0))) = 0);
END_ENTITY; -- assembly_group_component_definition
(*

```

Formal propositions:

WR1: The `assembly_group_component_definition` shall be related to at least one `component_definition` by a `product_definition_relationship` with a name of 'group component' or it shall be a definition for an `assembly_component_group_shape_aspect`.

WR2: If the `assembly_group_component_definition` is referenced by an `assembly_joint` through the `of_shape` attribute, at least one of the members of `component_definition` that are related by the `relating_shape_aspect` of that `assembly_joint` shall be related by members of `assembly_component_usage` to this `assembly_group_component_definition`.

WR3: If the `assembly_group_component_definition` is referenced by an `assembly_joint` through the `of_shape` attribute, each member of `component_definition` that are related by the `related_shape_aspect` of that `assembly_joint` shall be related by members of `assembly_component_usage` to this `assembly_group_component_definition`.

WR4: If the `assembly_group_component_definition` is referenced by an `assembly_joint` through the `of_shape.definition` attribute, the `component_definition` that is related by the `assembly_joint\shape_aspect_relationship.related_shape_aspect.of_shape.definition` attribute of that `assembly_joint` shall be positioned by a member of `component_location` which also references that `assembly_joint`.

5.2.3.1.32 `assembly_group_component_definition_placement_link`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `assembly_group_component_definition_placement_link` is a `product_definition` and a `product_definition_relationship` that implements the ARM concept of `Assembly_component_placement_link` (see 4.2.38)

EXPRESS specification:

```

*)
ENTITY assembly_group_component_definition_placement_link
  SUBTYPE OF (product_definition, product_definition_relationship);
WHERE

  WR1: SELF\product_definition_relationship.related_product_definition :<>:
        SELF\product_definition_relationship.relateing_product_definition;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.related_product_definition);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.relateing_product_definition);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\product_definition ||
        SELF\product_definition_relationship ||
        SELF\assembly_group_component_definition_placement_link)) = 0;

  WR5: SELF\product_definition.name = '';

  WR6: SELF\product_definition_relationship.name = '';

  WR7: SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')) = 1;
END_ENTITY; -- assembly_group_component_definition_placement_link
(*

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The assembly_group_component_definition_placement_link shall relate to only a component_definition in the related_product_definition.

WR3: The assembly_group_component_definition_placement_link shall relate to only a component_definition in the relating_product_definition.

WR4: The types of assembly_group_component_definition_placement_link shall be only assembly_group_component_definition_placement_link, product_definition, and product_definition_relationship.

WR5: The name attribute of the product_definition supertype shall be an empty string.

WR6: The name attribute of the product_definition_relationship supertype shall be an empty string.

WR7: Each `assembly_group_component_definition_placement_link` shall be referenced exactly once by a `product_definition_relationship` related `product_definition`.

5.2.3.1.33 `assembly_group_component_shape_aspect`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `assembly_group_component_shape_aspect` is a `component_shape_aspect` that implements the ARM concept of `Assembly_group_component` (see 4.2.40) .

EXPRESS specification:

```
*)
ENTITY assembly_group_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
WHERE

  WR1: (SIZEOF (QUERY (gc <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'group component') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN
    TYPEOF (gc.related_shape_aspect))))
    >= 1);
END_ENTITY; -- assembly_group_component_shape_aspect
(*
```

Formal propositions:

WR1: The `assembly_group_component_shape_aspect` shall be related to at least one `component_shape_aspect` by a `shape_aspect_relationship` with a name of 'group component'.

5.2.3.1.34 `assembly_group_component_shape_aspect_placement_link`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `assembly_group_component_shape_aspect_placement_link` is a `shape_aspect` and a `shape_aspect_relationship` that implements the ARM concept of `Assembly_component_placement_link` (see 4.2.38)

EXPRESS specification:

```

*)
ENTITY assembly_group_component_shape_aspect_placement_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\assembly_group_component_shape_aspect_placement_link)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';

  WR7: SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')) = 1;
END_ENTITY; -- assembly_group_component_shape_aspect_placement_link
(*

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The assembly_group_component_shape_aspect_placement_link shall relate to only a component_shape_aspect in the related_shape_aspect.

WR3: The assembly_group_component_shape_aspect_placement_link shall relate to only a component_shape_aspect in the relating_shape_aspect.

WR4: The types of assembly_group_component_shape_aspect_placement_link shall be only assembly_group_component_shape_aspect_placement_link, shape_aspect, and shape_aspect_relationship.

WR5: The name attribute of the shape_aspect supertype shall be an empty string.

WR6: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

WR7: Each `assembly_group_component_shape_aspect_placement_link` shall be referenced exactly once by a `shape_aspect_relationship` related `shape_aspect`.

5.2.3.1.35 `assembly_joint`

An `assembly_joint` is a `shape_aspect_relationship` and a `shape_aspect` that is the result of a physical connection between two components in the assembly. The order shall be inferred from the use of relating and related attributes on `shape_aspect_relationship`. An `assembly_joint` implements the ARM concepts of:

- `Assembly_joint` (see 4.2.41);
- `Assembled_with_bonding` (see 4.2.30);
- `Assembled_with_fasteners` (see 4.2.31).

EXPRESS specification:

```
*)
ENTITY assembly_joint
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
WHERE

  WR1: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (SELF.relating_shape_aspect)) AND
    (SELF.relating_shape_aspect\shape_aspect.description IN
    ['assembly module component terminal',
    'bare die component terminal',
    'interconnect component join terminal',
    'interconnect module component terminal',
    'package terminal occurrence',
    'packaged component join terminal'])) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE' IN
    TYPEOF (SELF.relating_shape_aspect)) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_INTERFACE_TERMINAL' IN
    TYPEOF (SELF.relating_shape_aspect)) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_MOUNTING_FEATURE' IN
    TYPEOF (SELF.relating_shape_aspect));
```

```

WR2: ((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL' IN
TYPEOF (SELF.related_shape_aspect)) AND
(SELF.related_shape_aspect\shape_aspect.description IN
['assembly module component terminal',
'bare die component terminal',
'interconnect component join terminal',
'interconnect module component terminal',
'package terminal occurrence',
'packaged component join terminal'])) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE' IN
TYPEOF (SELF.relateing_shape_aspect)) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_INTERFACE_TERMINAL' IN
TYPEOF (SELF.related_shape_aspect)) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_MOUNTING_FEATURE' IN
TYPEOF (SELF.relateing_shape_aspect)));

WR3: (NOT (SELF\shape_aspect_relationship.name =
'assembled with bonding')) OR
(SIZEOF (QUERY (ddu <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'default definition usage') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_BOND_DEFINITION' IN
TYPEOF (ddu.relateing_shape_aspect))) = 1));

WR4: (NOT (SELF\shape_aspect_relationship.name =
'assembled with bonding')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(pdr.used_representation\representation.name = 'bond assembly position') AND
(SIZEOF (QUERY (it <* pdr.used_representation.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MAPPED_ITEM' IN TYPEOF (it)) AND
(it\representation_item.name = 'assembly position')))) = 1))) = 1)) <= 1));

WR5: (NOT (SELF\shape_aspect_relationship.name =
'assembled with fasteners')) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_GROUP_COMPONENT_DEFINITION' IN
TYPEOF (SELF.of_shape.definition));

```

```

WR6: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_GROUP_COMPONENT_DEFINITION' IN
      TYPEOF (SELF.of_shape.definition)) OR
      is_assembly_module_design(SELF.of_shape.definition);

WR7: (NOT (SELF\shape_aspect_relationship.name =
      'assembled with bonding')) OR
      (SIZEOF (QUERY (ddu <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'default definition usage') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_BOND_DEFINITION' IN
      TYPEOF (ddu.relate_shape_aspect)) AND
      (SIZEOF (QUERY (sar <* USEDIN (ddu.relate_shape_aspect,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'bonded feature 1' )) = 1)
      AND (SIZEOF (QUERY (sar <* USEDIN (ddu.relate_shape_aspect,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'bonded feature 1' )) = 1)
      )) = 1);

WR8: acyclic_shape_aspect_relationship(SELF,
      [SELF\shape_aspect_relationship.related_shape_aspect],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_JOINT');
END_ENTITY; -- assembly_joint
( *
```

Formal propositions:

WR1: The assembly_joint shall have a relating_shape_aspect that references either a component_terminal with a description of 'assembly module component terminal', 'bare die component terminal', 'interconnect component join terminal', 'interconnect module component terminal', 'package terminal occurrence', or 'packaged component join terminal', or a component_interface_terminal or a component_mounting_feature or an interconnect_module_component_surface_feature.

WR2: The assembly_joint shall have a related_shape_aspect that references either a component_terminal with a description of 'assembly module component terminal', 'bare die component terminal', 'interconnect component join terminal', 'interconnect module component terminal', 'package terminal occurrence', or 'packaged component join terminal', or a component_interface_terminal or a component_mounting_feature or an interconnect_module_component_surface_feature.

WR3: If the name of the shape_aspect_relationship of the assembly_joint is 'assembled with bonding', it shall be related to exactly one assembly_bond_definition as the 'default definition usage'.

WR4: If the name of the shape_aspect_relationship of the assembly_joint is 'assembled with bonding', it shall have at most one property_definition that is associated with exactly one representation with a name of 'bond assembly position' that contains exactly one mapped_item with the name 'assembly position'.

WR5: If the name of the shape_aspect_relationship of the assembly_joint is 'assembled with fasteners', it shall be an aspect of the shape of an assembly_group_component_definition.

WR6: The assembly_joint is shall be an aspect of the shape of a physical_unit that is an assembly module design or of an assembly_group_component_definition.

WR7: If the name of the shape_aspect_relationship of the assembly_joint is 'assembled with bonding', its bonded features shall be defined by the referenced features.

WR8: The members of assembly_joint shall form a directed acyclic graph with the relating_shape_aspect, related_shape_aspect attributes of assembly_joint.

5.2.3.1.36 assembly_module_interface_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An assembly_module_interface_terminal is an assembly_module_terminal that implement the ARM concept of Assembly_module_interface_terminal (see 4.2.46)

EXPRESS specification:

```
*)
ENTITY assembly_module_interface_terminal
  SUBTYPE OF (assembly_module_terminal);
END_ENTITY; -- assembly_module_interface_terminal
(*
```

5.2.3.1.37 assembly_module_join_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An assembly_module_join_terminal is an assembly_module_terminal that implement the ARM concept of Assembly_module_join_terminal (see 4.2.47)

EXPRESS specification:

```
*)
ENTITY assembly_module_join_terminal
  SUBTYPE OF (assembly_module_terminal);
END_ENTITY; -- assembly_module_join_terminal
(*
```

5.2.3.1.38 assembly_module_macro_component_join_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An assembly_module_macro_component_join_terminal is a shape_aspect that implements the ARM concept of Assembly_module_macro_component_join_terminal (see 4.2.49)

EXPRESS specification:

```
*)
ENTITY assembly_module_macro_component_join_terminal
  SUBTYPE OF (component_terminal);
WHERE

  WR1: NOT (SELF\shape_aspect.description IN
    ['bare die component terminal',
     'component termination passage join terminal',
     'conductive interconnect element terminal',
     'interconnect component join terminal',
     'interconnect module component terminal', 'land join terminal',
     'minimally defined component terminal',
     'non functional land join terminal',
     'packaged component join terminal',
     'printed component join terminal', 'package terminal occurrence',
     'via terminal']);
END_ENTITY; -- assembly_module_macro_component_join_terminal
(*
```

Formal propositions:

WR1: The assembly_module_macro_component_join_terminal shall not have a description of 'bare die component terminal', 'component termination passage join terminal', 'conductive interconnect element terminal', 'interconnect component join terminal', 'interconnect module component terminal', 'land join terminal', 'minimally defined component terminal', 'non functional land join terminal', 'packaged component join terminal', 'printed component join terminal', 'package terminal occurrence', or 'via terminal'.

5.2.3.1.39 assembly_module_terminal

An assembly_module_terminal is a shape_aspect that implement the ARM concepts of:

- Assembly_module_terminal (see 4.2.50);
- Part_terminal (see 4.2.630);
- Pca_terminal (see 4.2.640).

EXPRESS specification:

```

*)
ENTITY assembly_module_terminal
  SUPERTYPE OF (ONEOF(
    assembly_module_interface_terminal,
    assembly_module_join_terminal))
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: EXISTS(SELF.of_shape.definition\product_definition.name);

  WR2: NOT EXISTS(SELF.of_shape.definition\product_definition.name) OR
    (SELF.of_shape.definition\product_definition.name = 'assembly module');

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation))) >= 1))) = 0;

  WR4: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'member connected terminal') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION'
    IN TYPEOF (mct.relate_shape_aspect))) <= 1;

  WR5: (NOT (SELF\shape_aspect.description = 'pca terminal')) OR
    ((SIZEOF(TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_ASSEMBLY_DEFINITION']) >=1)
    AND (SELF.of_shape.definition\product_definition.
    frame_of_reference.name = 'physical design usage'));

  WR6: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar.related_shape_aspect\shape_aspect.description = 'connection zone')) = 1;

```

```

WR7: sizeof (QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar.related_shape_aspect\shape_aspect.description = 'interface terminal')) <= 1;
END_ENTITY; -- assembly_module_terminal
(*

```

Formal propositions:

WR1: The product_definition name shall exist.

WR2: The assembly_module_terminal shall be the shape_aspect of a product_definition of a product that has a name of 'assembly module'.

WR3: The assembly_module_terminal shall have at least one shape_representation.

WR4: The assembly_module_terminal shall be related to at most one part_connected_terminals_definition as a 'member connected terminal'.

WR5: If the assembly_module_terminal has a description of 'pca terminal', it shall be an aspect of the shape of an assembly_definition, externally_defined_assembly_definition, or library_defined_assembly_definition that has a frame_of_reference name 'physical design usage'.

WR6: The assembly_module_terminal shall be related to exactly one shape_aspect with a description of 'connection zone'.

WR7: The assembly_module_terminal shall be related to at most one shape_aspect with a description of 'interface terminal'.

Informal propositions:

IP1: The assembly_module_terminal shall be related to at most one assembly_module_usage_view_connector_relationship.

5.2.3.1.40 assembly_module_usage_view_connector_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A assembly_module_usage_view_connector_relationship is a type of mapped_item that implements the ARM concept of assembly_module_usage_view_connector_relationship (see 4.2.52).

EXPRESS specification:

```

*)
ENTITY assembly_module_usage_view_connector_relationship
  SUBTYPE OF (mapped_item);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_3D' IN TYPEOF (SELF\mapped_item.mapping_target);

  WR2: SELF\mapped_item.mapping_source.mapped_representation\representation.name =
        '3d bound volume shape';

  WR3: SIZEOF (QUERY (pdr <* USEDIN
    ( SELF\mapped_item.mapping_source.mapped_representation,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_CONNECTOR',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PACKAGED_CONNECTOR',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PACKAGED_CONNECTOR'] *
      TYPEOF (pdr.definition.definition) ) = 1) )) = 1;

  WR4: SIZEOF (QUERY (r <* USEDIN ( SELF\representation_item,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') |
    SIZEOF (QUERY (pdr <* USEDIN (r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION' IN TYPEOF(pdr.definition.definition))
      AND
      (pdr.definition.definition.
        frame_of_reference\application_context_element.name =
        'physical design usage')
      AND
      (pdr.definition.definition\product_definition.
        name = 'assembly module')
      )) = 1 )) = 1;
END_ENTITY; -- assembly_module_usage_view_connector_relationship
(

```

Formal propositions:

WR1: The assembly_module_usage_view_connector_relationship shall have an axis2_placement_3d as a placement.

WR2: The assembly_module_usage_view_connector_relationship shall place a 3d representation.

WR3: The assembly_module_usage_view_connector_relationship shall place a 3d representation of a packaged_connector, externally_defined_packaged_connector, or a library_defined_packaged_connector.

WR4: The assembly_module_usage_view_connector_relationship shall place the 3d representation into a representation of a product_definition that is an 'assembly module'.

5.2.3.1.41 auxiliary_characteristic_dimension_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An auxiliary_characteristic_dimension_representation is a dimensional_characteristic_representation that implements the ARM concept of Dimensional_characteristic_representation (see 4.2.299) where the ARM attribute "auxiliary_dimension_representation" has a value of TRUE.

EXPRESS specification:

```
*)
ENTITY auxiliary_characteristic_dimension_representation
  SUBTYPE OF (dimensional_characteristic_representation);
END_ENTITY; -- auxiliary_characteristic_dimension_representation
(*
```

5.2.3.1.42 bare_die

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die is a physical_unit that implements the ARM concept of Bare_die (see 4.2.59).

EXPRESS specification:

```
*)
ENTITY bare_die
  SUBTYPE OF (physical_unit);
WHERE
```

```

WR1: sizeof (QUERY (ifdu <* QUERY (pdr <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRODUCT_DEFINITION_RELATIONSHIP.' +
' RELATED_PRODUCT_DEFINITION') |
pdr\property_definition_relationship.name = 'implemented function') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' FUNCTIONAL_UNIT' IN
typeof (ifdu.relying_product_definition)) AND
(ifdu.relying_product_definition.frame_of_reference.name =
'functional design usage')))) = 1;

WR2: sizeof (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
sizeof (QUERY (dut <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'device unit technology') |
dut.relying_property_definition\property_definition.name =
'unit technology')) = 1)) = 1;

WR3: sizeof (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (sizeof (QUERY (sa <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT.OF_SHAPE') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' BARE_DIE_TERMINAL' IN
typeof (sa))) >= 2))) = 0;
END_ENTITY; -- bare_die
( *

```

Formal propositions:

WR1: The bare_die shall be related to exactly one functional_unit with a frame_of_reference name of 'functional design usage' as the 'implemented function'.

WR2: The bare_die shall have exactly one property_definition that is related to exactly one property_definition with a name of 'unit technology' as the 'device unit technology'.

WR3: The bare_die shall have at least two shape_aspects that are bare_die_terminals.

Informal propositions:

IP1: The bare_die shall have exactly one representation with the name of material property data that shall be related to cartesian_points with names of 'least material condition centroid location' and 'maximum material condition centroid location'.

5.2.3.1.43 bare_die_bottom_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_bottom_surface is a shape_aspect that implements the ARM concept of Bare_die_bottom_surface (see 4.2.60).

EXPRESS specification:

```

*)
ENTITY bare_die_bottom_surface
  SUBTYPE OF (bare_die_surface);
WHERE

  WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_BARE_DIE']) = 1;

  WR2: SELF\shape_aspect.product_definitional;

  WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\bare_die_bottom_surface ||
    SELF\part_mounting_feature ||
    SELF\bare_die_surface)) = 0;
END_ENTITY; -- bare_die_bottom_surface
(

```

Formal propositions:

WR1: The bare_die_bottom_surface shall be the aspect of the shape of a bare_die, externally_defined_bare_die, or library_defined_bare_die.

WR2: The bare_die_bottom_surface shall be on the boundary of the product.

WR3: The bare_die_bottom_surface shall be no other subtype of shape_aspect.

5.2.3.1.44 bare_die_edge_segment_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_edge_segment_surface is a shape_aspect and shape_aspect_relationship that implements the ARM concept of Bare_die_edge_segment_surface (see 4.2.63).

EXPRESS specification:

```

*)
ENTITY bare_die_edge_segment_surface
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_SEGMENT_VERTEX' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_SEGMENT_VERTEX' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);

  WR3: SELF\shape_aspect_relationship.relying_shape_aspect :<>:
        SELF\shape_aspect_relationship.related_shape_aspect;

  WR4: SIZEOF (QUERY (ce < * QUERY (sar < * USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'composed surface') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_EDGE_SURFACE' IN TYPEOF
        (ce.relying_shape_aspect) ))) = 1;

  WR5: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\bare_die_edge_segment_surface)) = 0;
END_ENTITY; -- bare_die_edge_segment_surface
( *

```

Formal propositions:

WR1: The bare_die_edge_segment_surface shall be related to exactly one edge_segment_vertex through a shape_aspect_relationship related attribute.

WR2: The bare_die_edge_segment_surface shall be related to exactly one edge_segment_vertex through a shape_aspect_relationship relating attribute.

WR3: The edge_segment_vertex instances referenced shall not be the same instance of edge_segment_vertex.

WR4: The bare_die_edge_segment_surface shall be related to exactly one bare_die_edge_surface as the 'composed surface'.

WR5: The bare_die_edge_segment_surface shall not be in a complex instance other than with its immediate supertypes.

5.2.3.1.45 bare_die_edge_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_edge_surface is a shape_aspect that implements the ARM concept of Bare_die_edge_surface (see 4.2.64).

EXPRESS specification:

```

*)
ENTITY bare_die_edge_surface
  SUBTYPE OF (bare_die_surface);
WHERE

  WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_BARE_DIE']) = 1;

  WR2: SELF\shape_aspect.product_definitional;

  WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\part_mounting_feature ||
    SELF\bare_die_edge_surface ||
    SELF\bare_die_surface)) = 0;
END_ENTITY; -- bare_die_edge_surface
(

```

Formal propositions:

WR1: The bare_die_edge_surface shall be the aspect of the shape of a bare_die, externally_defined_bare_die, or library_defined_bare_die.

WR2: The bare_die_edge_surface shall be on the boundary of the product.

WR3: The bare_die_edge_surface shall be no other subtype of shape_aspect.

5.2.3.1.46 bare_die_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_surface is a shape_aspect that implements the ARM concept of Bare_die_surface (see 4.2.65).

EXPRESS specification:

```

*)
ENTITY bare_die_surface
  ABSTRACT SUPERTYPE OF (ONEOF( bare_die_bottom_surface,
                                bare_die_top_surface,
                                bare_die_edge_surface))
  SUBTYPE OF (part_mounting_feature);
WHERE

  WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_BARE_DIE']) = 1;

  WR2: SELF\shape_aspect.product_definitional;
END_ENTITY; -- bare_die_surface
(*

```

Formal propositions:

WR1: The bare_die_surface shall be the aspect of the shape of a bare_die, externally_defined_bare_die, or library_defined_bare_die.

WR2: The bare_die_surface shall be on the boundary of the product.

5.2.3.1.47 bare_die_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_terminal is a minimally_defined_bare_die_terminal that implements the ARM concept of Bare_die_terminal (see 4.2.66).

EXPRESS specification:

```

*)
ENTITY bare_die_terminal
  SUBTYPE OF (minimally_defined_bare_die_terminal);
WHERE

```

```

WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE' IN
      TYPEOF (SELF.of_shape.definition);

WR2: SIZEOF (QUERY (eca <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'external connection area') |
      eca.related_shape_aspect\shape_aspect.description = 'connection zone')) = 1;

WR3: SIZEOF (USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MATERIAL_DESIGNATION.DEFINITIONS')) = 1;

WR4: SIZEOF (QUERY (eca <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'seating plane zone') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONNECTION_ZONE_INTERFACE_PLANE_RELATIONSHIP' IN
      TYPEOF (eca.related_shape_aspect))) <= 1;
END_ENTITY; -- bare_die_terminal
( *

```

Formal propositions:

WR1: The bare_die_terminal shall be a terminal of a bare_die.

WR2: The bare_die_terminal shall relate to exactly one shape_aspect that has a description of 'connection zone' as the 'external connection area'.

WR3: The bare_die_terminal shall have exactly one material designation.

WR4: The bare_die_terminal shall have no more than one 'seating plane zone'.

5.2.3.1.48 bare_die_top_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bare_die_top_surface is a shape_aspect that implements the ARM concept of Bare_die_top_surface (see 4.2.68).

EXPRESS specification:

```

*)
ENTITY bare_die_top_surface
  SUBTYPE OF (bare_die_surface);
WHERE

  WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_BARE_DIE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_BARE_DIE']) = 1;

  WR2: SELF\shape_aspect.product_definitional;

  WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\part_mounting_feature ||
    SELF\bare_die_top_surface ||
    SELF\bare_die_surface)) = 0;
END_ENTITY; -- bare_die_top_surface
(

```

Formal propositions:

WR1: The bare_die_top_surface shall be the aspect of the shape of a bare_die, externally_defined_bare_die, or library_defined_bare_die.

WR2: The bare_die_top_surface shall be on the boundary of the product.

WR3: The bare_die_top_surface shall be no other subtype of shape_aspect.

5.2.3.1.49 bond_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bond_category is a group and an external_definition that implements the ARM concept of External_definition (see 4.2.361) in the role of bond category.

EXPRESS specification:

```

*)
ENTITY bond_category
  SUBTYPE OF (group, external_definition);
WHERE

```

```

    WR1: SIZEOF (USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;
END_ENTITY; -- bond_category
(*

```

Formal propositions:

WR1: The bond_category shall be referenced by exactly one document.

5.2.3.1.50 bus_element_link

A bus_element_link is a product_definition and a product_definition_relationship that implements the ARM concept of Bus_element_link (see 4.2.82)

EXPRESS specification:

```

*)
ENTITY bus_element_link
    SUBTYPE OF (product_definition, product_definition_relationship);
UNIQUE
    UR1: SELF\product_definition_relationship.related_product_definition,
        SELF\product_definition_relationship.relying_product_definition;
WHERE

    WR1: SELF\product_definition_relationship.related_product_definition :<>:
        SELF\product_definition_relationship.relying_product_definition;

    WR2: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.related_product_definition)) OR
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NETWORK_NODE_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.related_product_definition));

    WR3: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.relying_product_definition)) OR
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NETWORK_NODE_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.relying_product_definition));

    WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\product_definition ||
        SELF\product_definition_relationship ||
        SELF\bus_element_link)) = 0;

```

```

WR5: NOT EXISTS(SELF\product_definition.name) OR
      (SELF\product_definition.name = '');

WR6: SELF\product_definition_relationship.name = '';

WR7: sizeof(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')) >= 1;
END_ENTITY; -- bus_element_link
(*)

```

Formal propositions:

UR1: The combination of related, relating attributes inherited from product_definition_relationship be unique within a population of bus_element_link.

WR1: The related element shall not be the relating element.

WR2: The bus_element_link shall relate to only a bus_structural_definition or a network_node_definition in the related_product_definition.

WR3: The bus_element_link shall relate to only a bus_structural_definition or a network_node_definition in the relating_product_definition.

WR4: The types of bus_element_link shall be only bus_element_link, product_definition, and product_definition_relationship.

WR5: The name attribute of the product_definition supertype shall be an empty string.

WR6: The name attribute of the product_definition_relationship supertype shall be an empty string.

WR7: Each bus_element_link shall be referenced at least once by a product_definition_relationship related_product_definition.

5.2.3.1.51 bus_structural_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A bus_structural_definition is a product_definition that implements the ARM concept of Bus_structural_definition (see 4.2.83).

EXPRESS specification:

```

*)
ENTITY bus_structural_definition
  SUBTYPE OF (product_definition);
WHERE

```

```

WR1: sizeof (QUERY (bce <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'bus composition') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BUS_ELEMENT_LINK' IN
    TYPEOF (bce.related_product_definition))) >= 1;

WR2: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    sizeof (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name =
    'functional unit network terminal definition bus assignment'))
    <= 1)) <= 1;

WR3: consistent_bus_structural_definition(bag_to_set(QUERY( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BUS_ELEMENT_LINK' IN TYPEOF(pdr.related_product_definition))));
END_ENTITY; -- bus_structural_definition
(*

```

Formal propositions:

WR1: The bus_structural_definition shall relate to at least one bus_element_link.

WR2: The bus_structural_definition shall have at most one property_definition that relates to another property_definition to specify the 'functional unit network terminal definition bus assignment'.

WR3: The bus_structural_definition shall be a connected structure and shall not have any vertex with a degree greater than 2. The number of vertices shall equal the number of links + 1.

5.2.3.1.52 cable_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A cable_component is a component_definition that implements the ARM concept of Cable_component (see 4.2.84)

EXPRESS specification:

```

*)
ENTITY cable_component
  SUBTYPE OF (component_definition);
WHERE

  WR1: SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'instantiated part') |
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT'] *
    TYPEOF (ip.relatng_product_definition)) = 1) AND
    (ip.relatng_product_definition.frame_of_reference.name =
    'physical design usage')))) = 1;

  WR2: SELF\product_definition.description :<>: 'laminate component';
END_ENTITY; -- cable_component
( *

```

Formal propositions:

WR1: The cable_component shall be related, as the 'instantiated part', to exactly one physical_unit that is in the 'physical design usage' product_definition_context.

WR2: The cable_component shall not have a description of 'laminate component'.

5.2.3.1.53 capacitance_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A capacitance_measure_with_unit is a measure_with_unit that implements the ARM concept of Capacitance_measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY capacitance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\capacitance_measure_with_unit))) = 0;

```

```

WR2: SELF\measure_with_unit.unit_component\si_unit.name = farad;

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
      TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- capacitance_measure_with_unit
(*)

```

Formal propositions:

WR1: The capacitance_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The capacitance_measure_with_unit shall reference an si_unit with a name of farad.

WR3: The unit_component shall be an si_unit.

5.2.3.1.54 change

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A change is an action_assignment that points to a set of **work_item**.

EXPRESS specification:

```

*)
ENTITY change
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF work_item;
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DIRECTED_ACTION'
        IN TYPEOF (SELF.assigned_action);

  WR2: unique_version_change_order (SELF.assigned_action,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN');

END_ENTITY; -- change
(*)

```

Formal propositions:

WR1: The change shall assign a directed_action.

WR2: A single change shall not change more than one version of a single product, but may change more than one product_definition_formation if each product_definition_formation references a different product.

5.2.3.1.55 change_from_design_object_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A change_from_design_object_assignment is an action_assignment that implements the ARM concepts of:

- Change_design_object_management_relationship (see 4.2.95);
- Component_swap_specification (see 4.2.136);
- Gate_path_swap_specification (see 4.2.415);
- Terminal_swap_specification (see 4.2.891).

EXPRESS specification:

```

*)
ENTITY change_from_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(aa <* USEDIN(SELF\action_assignment.assigned_action,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ACTION_ASSIGNMENT.ASSIGNED_ACTION')
    | 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CHANGE_TO_DESIGN_OBJECT_ASSIGNMENT' IN TYPEOF(aa))) >= 1;
END_ENTITY; -- change_from_design_object_assignment
( *
```

Formal propositions:

WR1: SELF\action_assignment.assigned_action shall also be a SELF\action_assignment.assigned_action for a change_to_design_object_assignment.

Informal propositions:

IP1: The change_from_design_object_assignment shall have exactly one item that is a managed_design_object that is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design that is not an item of a action_assignment with the same assigned_action.

5.2.3.1.56 change_from_design_object_request_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A change_from_design_object_request_assignment is an action_request_assignment that implements the ARM concepts of:

- Change_design_object_management_relationship (see 4.2.95);
- Change_request (see 4.2.97);
- Component_swap_specification (see 4.2.136);
- Gate_path_swap_specification (see 4.2.415);
- Terminal_swap_specification (see 4.2.891).

EXPRESS specification:

```

*)
ENTITY change_from_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(aa <*
    USEDIN(SELF\action_request_assignment.assigned_action_request,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST')
    | 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CHANGE_TO_DESIGN_OBJECT_REQUEST_ASSIGNMENT' IN TYPEOF(aa))) >= 1;

END_ENTITY; -- change_from_design_object_request_assignment
( *
```

Formal propositions:

WR1: SELF\action_request_assignment.assigned_action_request shall also be a SELF\action_request_assignment.assigned_action_request for a change_to_design_object_request_assignment.

Informal propositions:

IP1: The `change_from_design_object_request_assignment` shall have exactly one item that is a managed_design_object that is not referenced by another `product_definition_relationship` with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design that is not an item of a `action_request_assignment` with the same `assigned_action_request`.

5.2.3.1.57 change_request

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `change_request` is an `action_request_assignment` that implements the ARM concept of `Change_request` (see 4.2.97).

EXPRESS specification:

```
*)
ENTITY change_request
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF change_request_item;
END_ENTITY; -- change_request
(*
```

5.2.3.1.58 change_to_design_object_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `change_to_design_object_assignment` is an `action_assignment` that implements the ARM concepts of:

- `Change_design_object_management_relationship` (see 4.2.95);
- `Component_swap_specification` (see 4.2.136);
- `Gate_path_swap_specification` (see 4.2.415);
- `Terminal_swap_specification` (see 4.2.891).

EXPRESS specification:

```
*)
ENTITY change_to_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE
```

```

WR1: sizeof(QUERY(aa <* USEDIN(SELF\action_assignment.assigned_action,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ACTION_ASSIGNMENT.ASSIGNED_ACTION')
| 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CHANGE_FROM_DESIGN_OBJECT_ASSIGNMENT' IN TYPEOF(aa))) >= 1;

WR2: sizeof(QUERY ( pdr <* SELF.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(pdr)) AND
(pdr\product_definition_relationship.name = 'design object change')))) = 1;

END_ENTITY; -- change_to_design_object_assignment
( *
```

Formal propositions:

WR1: SELF\action_assignment.assigned_action shall also be a SELF\action_assignment.assigned_action for a change_from_design_object_assignment.

WR2: The change_to_design_object_assignment shall have exactly one item that is a product_definition_relationship with a name of 'design object change'.

Informal propositions:

IP1: The change_to_design_object_assignment shall have exactly one item that is a managed_design_object that is not a product_definition_relationship with a name of 'design object change' and is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design that is not an item of a action_assignment with the same assigned_action.

5.2.3.1.59 change_to_design_object_request_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A change_to_design_object_request_assignment is an action_request_assignment that implements the ARM concepts of:

- Change_design_object_management_relationship (see 4.2.95);
- Component_swap_specification (see 4.2.136);
- Change_request (see 4.2.97);
- Gate_path_swap_specification (see 4.2.415);
- Terminal_swap_specification (see 4.2.891).

EXPRESS specification:

```

*)
ENTITY change_to_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(aa <*
    USEDIN(SELF\action_request_assignment.assigned_action_request,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST')
    | 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CHANGE_FROM_DESIGN_OBJECT_REQUEST_ASSIGNMENT'
    IN TYPEOF(aa))) >= 1;

  WR2: SIZEOF(QUERY ( pdr <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(pdr)) AND
    (pdr\product_definition_relationship.name = 'design object change')) = 1;
END_ENTITY; -- change_to_design_object_request_assignment
(*

```

Formal propositions:

WR1: SELF\action_request_assignment.assigned_action_request shall also be a SELF\action_request_assignment.assigned_action_request for a change_from_design_object_request_assignment.

WR2: The change_to_design_object_request_assignment shall have exactly one item that is a product_definition_relationship with a name of 'design object change'.

Informal propositions:

IP1: The change_to_design_object_request_assignment shall have exactly one item that is a managed_design_object that is not a product_definition_relationship with a name of 'design object change' and is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design that is not an item of a action_request_assignment with the same assigned_action_request.

5.2.3.1.60 characteristic_type

A characteristic_type is a group that specifies a boolean property, a numeric property whose units are in accordance with ISO 10303-41, a compound characteristic, a logical property, a string property, or an externally defined property for a characteristic.

EXPRESS specification:

```
*)
ENTITY characteristic_type
  SUBTYPE OF (group);
WHERE

  WR1: SELF\group.name IN [
    'string property',
    'logical property',
    'boolean property',
    'numeric property',
    'externally defined property',
    'compound characteristic'
  ];
END_ENTITY; -- characteristic_type
(*
```

Formal propositions:

WR1: A `characteristic_type` shall have a name that is 'string property', 'logical property', 'boolean property', 'numeric property', 'externally defined property' or 'compound characteristic'.

5.2.3.1.61 characterized_product_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `characterized_product_category` is a `characterized_object` and a `product_category` that implements the ARM concept of `Ee_product_category` (see 4.2.320).

EXPRESS specification:

```
*)
ENTITY characterized_product_category
  SUBTYPE OF (characterized_object, product_category);
WHERE
```

```

WR1:  SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (pd\property_definition.description = 'product category values') AND
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PARAMETER_ASSIGNMENT_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)))) >= 1 ))) <= 1;

WR2:  SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    ((pd\property_definition.description = 'product category parameters') AND
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF (QUERY (ri <* pdr.used_representation.items
    | (NOT (SIZEOF (TYPEOF (ri) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODEL_PARAMETER']) =1))) =0))) >= 1 )))) <= 1;

WR3:  SELF\characterized_object.name = SELF\product_category.name;

WR4:  SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;

```

```

END_ENTITY; -- characterized_product_category
(*)

```

Formal propositions:

WR1: The characterized_product_category shall be referenced by no more than one property_definition that has a description of 'product category values' that shall reference one or more parameter_assignment_representation.

WR2: The characterized_product_category shall be referenced by no more than one property_definition that has a description of 'product category parameters' that shall reference one or more model_parameter.

WR3: The name attribute inherited from product_category shall be the same value as the name attribute inherited from characterized_object.

WR4: The characterized_product_category shall be referenced by exactly one applied_document_reference.

NOTE The RULE characterized_product_category_unique_constraint also constrains this entity.

5.2.3.1.62 circular_runout_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `circular_runout_tolerance` is a `geometric_tolerance_with_specified_datum_system` that implements the ARM concept of `Circular_runout_tolerance` (see 4.2.102).

EXPRESS specification:

```
*)
ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'circular runout';

  WR2: NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF));

  WR3: SIZEOF (QUERY (pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (pdr\property_definition_relationship.name = 'referenced datum system definition')
    (pdr.related_property_definition\property_definition.description =
    'datum system property without material conditions')))) = 1;
END_ENTITY; -- circular_runout_tolerance
(*
```

Formal propositions:

WR1: The name for the `circular_runout_tolerance` shall be 'circular runout'.

WR2: The `circular_runout_tolerance` shall not be a `modified_geometric_tolerance`.

WR3: The `circular_runout_tolerance` shall relate to exactly one `property_definition` that has the description of 'datum system property without material conditions' to specify the 'referenced datum system definition'.

Informal propositions:

IP1: The `toleranced_shape_aspect` of the `circular_runout_tolerance` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP2: The `circular_runout_tolerance` shall have at most two `physical_unit_datum` in its referenced `datum_system`.

IP3: The `circular_runout_tolerance` shall have at least one `physical_unit_datum` in its referenced `datum_system` that has description of 'axis'.

IP4: The `tolerance_zone_boundary` of the `circular_runout_tolerance` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

5.2.3.1.63 **component_definition**

A `component_definition` is a `product_definition` that represents a product that is used in an assembly. The `component_definition` entity implements the ARM concepts of:

- `Array_placement_group` (see 4.2.29);
- `Assembly_component` (see 4.2.34);
- `Assembly_group_component` (see 4.2.40);
- `Assembly_module_component` (see 4.2.44);
- `Assembly_module_macro_component` (see 4.2.48);
- `Bare_die_component` (see 4.2.61);
- `Cable_component` (see 4.2.84);
- `Interconnect_module_component` (see 4.2.442);
- `Interconnect_module_macro_component` (see 4.2.455);
- `Interface_component` (see 4.2.468);
- `Interfaced_group_component` (see 4.2.473);
- `Mating_connector_component` (see 4.2.538);
- `Physical_component` (see 4.2.648);
- `Physical_laminate_component` (see 4.2.659);
- `Routed_interconnect_component` (see 4.2.777);
- `Routed_physical_component` (see 4.2.778);
- `Thermal_component` (see 4.2.900).

EXPRESS specification:

```

*)
ENTITY component_definition
  SUPERTYPE OF (ONEOF (printed_component, packaged_component, routed_physical_component)
    ANDOR (thermal_component) ANDOR (cable_component))
  SUBTYPE OF (product_definition);
WHERE

  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)) |
    sr_pdr.used_representation\representation.name = 'planar projected shape'))
    <= 1))) = 0;

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)) |
    sr_pdr.used_representation\representation.name = '3d bound volume shape'))
    <= 1))) = 0;

  WR3: (NOT (SELF\product_definition.description =
    'bare die component')) OR
    (SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'instantiated part') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_BARE_DIE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_BARE_DIE'] *
    TYPEOF (ip.relatng_product_definition)) = 1) AND
    (ip.relatng_product_definition.frame_of_reference.name =
    'physical design usage')) = 1);

```



```

WR4: NOT(is_assembly_module_occurrence(SELF))
OR
(SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'instantiated part') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT' IN
TYPEOF (ip.relying_product_definition)) AND
(ip.relying_product_definition.frame_of_reference.name =
'physical design usage') AND
(ip.relying_product_definition\product_definition.name =
'assembly module')
)) = 1);

WR5: NOT(is_assembly_module_occurrence(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL' IN
TYPEOF (sa)) AND
(sa\shape_aspect.description = 'assembly module component terminal'))
>= 2)) >= 1);

WR6: NOT(is_interconnect_module_occurrence(SELF))
OR
(SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'instantiated part') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT' IN
TYPEOF (ip.relying_product_definition)) AND
(ip.relying_product_definition.frame_of_reference.name =
'physical design usage') AND
(SELF\product_definition.name = 'interconnect module')
)) = 1);

```

```

WR7: (NOT (SELF\product_definition.description = 'mating connector')) OR
((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
TYPEOF (SELF)) AND
(SIZEOF (QUERY (doc <*
SELF\product_definition_with_associated_documents.documentation_ids |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EE_SPECIFICATION' IN
TYPEOF (doc))) = 1)) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pt_occ <* QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
sa\shape_aspect.description = 'part template occurrence') |
SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (pt_occ,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION' IN
TYPEOF (it.relateing_shape_aspect)) AND
(it.relateing_shape_aspect\shape_aspect.description =
'printed connector template')) = 1)) = 1))) = 0) OR
(SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'instantiated part') |
(SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] *
TYPEOF (ip.relateing_product_definition)) = 1) AND
(ip.relateing_product_definition\product_definition.description =
'packaged connector')) = 1));

WR8: (NOT (SELF\product_definition.description = 'mating connector')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'mating connector placement')) = 1)) = 1);

```

```

WR9: (NOT (SELF\product_definition.description = 'mating connector')) OR
      ((SIZEOF (QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NEXT_ASSEMBLY_USAGE_OCCURRENCE' IN TYPEOF(pdr))) = 0) AND
      (SIZEOF (QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NEXT_ASSEMBLY_USAGE_OCCURRENCE' IN TYPEOF(pdr))) = 0));

WR10: SIZEOF(QUERY(pdr <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  NOT(NOT(pdr\product_definition_relationship.name = 'instantiated part')
  OR
  (SELF.formation :=:
  pdr.relating_product_definition.formation))))
  = 0;

WR11: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_REPRESENTATION'
  IN TYPEOF (pdr.used_representation)) |
  (sr_pdr.used_representation\representation.name =
  'part template non planar 2d shape')
  OR
  (sr_pdr.used_representation\representation.name = 'non planar 2d shape')
  OR
  (sr_pdr.used_representation\representation.name =
  'open shell based surface'))))
  <= 1))) = 0;

WR12: NOT(SELF.frame_of_reference.name = 'layout occurrence') OR
      ((SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
        prpc\product_category.name = 'template model')) >= 1) OR
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_GROUP_COMPONENT_DEFINITION' IN TYPEOF(SELF)));

```

```

WR13: NOT((SELF.frame_of_reference.name = 'layout occurrence')
AND
NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_COMPONENT' IN TYPEOF(SELF))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
((SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT'] *
TYPEOF (sa)) >= 1))
)) = 1))) = 0);

```

```

WR14: NOT(is_assembly_module_macro_occurrence(SELF))
OR
(SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'design definition') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT' IN
TYPEOF (ip.relying_product_definition)) AND
(ip.relying_product_definition.frame_of_reference.name =
'physical design') AND
(ip.relying_product_definition\product_definition.name =
'assembly module')
)) = 1);

```

```

WR15: NOT(is_interconnect_module_macro_occurrence(SELF))
OR
(SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'design definition') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT' IN
TYPEOF (ip.relying_product_definition)) AND
(ip.relying_product_definition.frame_of_reference.name =
'physical design') AND
(ip.relying_product_definition\product_definition.name =
'interconnect module')
)) = 1);

```

```

WR16: SIZEOF(QUERY(pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
((pdr\product_definition_relationship.name = 'instantiated definition')
AND
(NOT(SELF.formation :=:
pdr.relatng_product_definition.formation))))))
= 0;

WR17: (NOT (SELF\product_definition.description =
'routed interconnect component')) OR
((SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'instantiated part') |
(SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT'] *
TYPEOF (ip.relatng_product_definition)) = 1) AND
(ip.relatng_product_definition.frame_of_reference.name =
'physical design usage')) = 1) AND
(SELF.frame_of_reference.name = 'physical occurrence'));
END_ENTITY; -- component_definition
(*

```

Formal propositions:

WR1: The component_definition shall have at most one shape_representation with a name of 'planar projected shape'.

WR2: The component_definition shall have at most one shape_representation with a name of '3d bound volume shape'.

WR3: If the component_definition has a description of 'bare die component', the component_definition shall be related, as the 'instantiated part', to exactly one bare_die, externally_defined_bare_die, or library_defined_bare_die; that is in the 'physical design usage' product_definition_context.

WR4: If the component_definition is in the 'physical occurrence' product_definition_context and has a name of 'assembly module', the component_definition shall be related, as the 'instantiated part', to exactly one physical_unit that is in the 'physical design usage' product_definition_context and has a name of 'assembly module'.

WR5: If the component_definition has a frame_of_reference name of 'physical occurrence' and it is associated with a product that is categorized as an 'assembly module', it shall have at least two shape_aspects that are component_terminals with a description of 'assembly module component terminal'.

WR6: If the component_definition is in the 'physical occurrence' product_definition_context and has a name of 'interconnect module', the component_definition shall be related, as the 'instantiated part', to exactly one physical_unit that is in the 'physical design usage' product_definition_context and has a name that is 'interconnect module'.

WR7: If the `component_definition` has a description of 'mating connector', it shall be of type `product_definition_with_associated_documents` that contains exactly one `ee_specification`, or it shall have exactly one `shape_aspect` with a description of 'part template occurrence' and is related to exactly one `part_template_definition` with a description of 'printed connector template' as the 'instantiated template', or it shall be related to exactly one `packaged_part`, `externally_defined_packaged_part`, or `library_defined_packaged_part` with a description of 'packaged connector' as the 'instantiated part'.

WR8: If the `component_definition` has a description of 'mating connector', it shall have exactly one `shape_representation` with a name of 'mating connector placement'.

WR9: If the `component_definition` has a description of 'mating connector', it shall not participate in a `next_assembly_usage_occurrence`.

WR10: If the `component_definition` is the `related_product_definition` of a `product_definition_relationship` that has a name of 'instantiated part', it shall have the same formation as the `relating_product_definition`.

WR11: The `component_definition` shall have at most one `shape_representation` with a name of 'part template non planar 2d shape' or of 'open shell based surface' or of 'non planar 2d shape'.

WR12: If the `component_definition` has a `frame_of_reference` with a name of 'layout occurrence', the `component_definition` shall be a definition of a product that is categorized as a 'template model'.

WR13: If the `component_definition` has a `frame_of_reference` with a name of 'layout occurrence', and if the `component_definition` is not a `printed_component`, the `component_definition` shall have exactly one `component_shape_aspect`.

WR14: If the `component_definition` is in the 'macro occurrence' `product_definition_context` and has a name of 'assembly module', the `component_definition` shall be related, by the 'design definition' relationship, to exactly one `physical_unit` that is in the 'physical design' `product_definition_context` and has a name of 'assembly module'.

WR15: If the `component_definition` is in the 'macro occurrence' `product_definition_context` and has a name of 'interconnect module', the `component_definition` shall be related, by the 'design definition' relationship, to exactly one `physical_unit` that is in the 'physical design' `product_definition_context` and has a name of 'interconnect module'.

WR16: If the `component_definition` is the `related_product_definition` of a `product_definition_relationship` that has a name of 'instantiated definition', it shall have the same formation as the `relating_product_definition`.

WR17: If the `component_definition` has a description of 'routed interconnect component', it shall be related to a `physical_unit` as an 'instantiated part' and shall be in the 'physical occurrence' context.

Informal propositions:

IP1: The assembly referenced by the assembly composition relationship that references this component definition shall be the same assembly as that referenced by the shape that is referenced by the component position that references that assembly composition relationship.

IP2: If the component_definition has a description of 'routed interconnect component', it shall have exactly one property_definition that is associated to exactly one representation.

5.2.3.1.64 component_feature

A component_feature is a shape_aspect. A component_feature implements the ARM concept of Component_feature (see 4.2.118) when no ARM subtype is implemented.

EXPRESS specification:

```

*)
ENTITY component_feature
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);

  WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'instantiated feature')) <= 1;
END_ENTITY; -- component_feature
(

```

Formal propositions:

WR1: The component_feature shall be an aspect of the shape of a component_definition.

WR2: The component_feature shall be related to at most one shape_aspect as an 'instantiated feature'.

5.2.3.1.65 component_feature_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A component_feature_relationship is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Component_feature_relationship (see 4.2.121)

EXPRESS specification:

```

*)
ENTITY component_feature_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relate_shape_aspect;

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\component_feature_relationship)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';
END_ENTITY; -- component_feature_relationship
(

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The types of component_feature_relationship shall be only component_feature_relationship, shape_aspect, and shape_aspect_relationship.

WR3: The name attribute of the shape_aspect supertype shall be an empty string.

WR4: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

5.2.3.1.66 component_functional_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A component_functional_terminal is a shape_aspect that implements the ARM concept of Functional_unit_terminal (see 4.2.407).

EXPRESS specification:

```

*)
ENTITY component_functional_terminal
  SUBTYPE OF (shape_aspect);
WHERE

```



```

WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_FUNCTIONAL_UNIT' IN
      TYPEOF (SELF.of_shape.definition);

WR2: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'instantiated terminal') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT_TERMINAL_DEFINITION'
      IN TYPEOF (it.relatng_shape_aspect))) = 1;

WR3: SIZEOF (QUERY (futba <* QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      pdr\property_definition_relationship.name = 'functional unit terminal bus assignment'
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BUS_STRUCTURAL_DEFINITION' IN
      TYPEOF (futba.relatng_property_definition.definition))) <= 1;

WR4: SIZEOF (QUERY (futna <* QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      pdr\property_definition_relationship.name = 'functional unit terminal node assignment'
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'NETWORK_NODE_DEFINITION' IN
      TYPEOF (futna.relatng_property_definition.definition))) <= 1;
END_ENTITY; -- component_functional_terminal
(*

```

Formal propositions:

WR1: The component_functional_terminal shall be an aspect of the shape of a component_functional_unit.

WR2: The component_functional_terminal shall be related to exactly one functional_unit_terminal_definition as the 'instantiated terminal'.

WR3: The component_functional_terminal shall be related to at most one property of a bus_structural_definition as the 'functional unit terminal bus assignment'.

WR4: The component_functional_terminal shall be related to at most one property of a network_node_definition as the 'functional unit terminal node assignment'.

5.2.3.1.67 component_functional_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A component_functional_unit is a product_definition that implements the ARM concept of Functional_unit (see 4.2.400).

EXPRESS specification:

```

*)
ENTITY component_functional_unit
  SUBTYPE OF (product_definition);
WHERE

  WR1: SELF.frame_of_reference.name = 'functional occurrence';

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_FUNCTIONAL_TERMINAL' IN
    TYPEOF (sa))) >= 1)) >= 1;

  WR3: SIZEOF (QUERY (ifu <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'instantiated functional unit') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' IN
    TYPEOF (ifu.relating_product_definition))) = 1;

  WR4: SIZEOF (QUERY (nc <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'network composition') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' IN
    TYPEOF (nc.relating_product_definition)) AND
    (nc.relating_product_definition.frame_of_reference.name =
    'functional network design')) = 1;
END_ENTITY; -- component_functional_unit
(

```

Formal propositions:

WR1: The component_functional_unit shall have the context element named 'functional occurrence'.

WR2: The component_functional_unit shall have at least one component_functional_terminal.

WR3: The component_functional_unit shall be related to exactly one functional_unit as the 'instantiated functional unit'.

WR4: The component_functional_unit shall be related to exactly one functional_unit that has a context element with a name of 'functional network design' as the 'network composition'.

5.2.3.1.68 component_interface_terminal

A component_interface_terminal is a shape_aspect that implements the ARM concepts of:

- Component_interface_terminal (see 4.2.125);
- Interconnect_component_interface_terminal (see 4.2.438);
- Packaged_connector_component_interface_terminal (see 4.2.589).

EXPRESS specification:

```

*)
ENTITY component_interface_terminal
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\shape_aspect.description IN [
    'interconnect component interface terminal',
    'packaged connector component interface terminal'];

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature')) <= 1;

  WR4: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'design usage')) <= 1;

  WR5: (NOT (SELF\shape_aspect.description =
    'packaged connector component interface terminal')) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERFACE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition));

  WR6: (NOT (SELF\shape_aspect.description =
    'packaged connector component interface terminal')) OR
    (SELF.of_shape.definition\product_definition.
    frame_of_reference.name =
    'physical occurrence');
```

```

WR7: (NOT (SELF\shape_aspect.description =
'interconnect component interface terminal')) OR
((SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
(sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) = 1)
AND
NOT(SELF.of_shape.definition\product_definition.description
IN ['assembly module component', 'bare die component',
'interconnect module component', 'lamine component',
'packaged component']));

WR8: (NOT (SELF\shape_aspect.description =
'packaged connector component interface terminal')) OR
(SELF.product_definitional);

WR9: (NOT (SELF\shape_aspect.description =
'packaged connector component interface terminal')) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_COMPONENT' IN
TYPEOF (SELF.of_shape.definition)) AND
(SELF.of_shape.definition\product_definition.description =
'packaged connector component') AND
(SELF.of_shape.definition\product_definition.
frame_of_reference.name =
'physical occurrence'));

WR10: (NOT (SELF\shape_aspect.description =
'packaged connector component interface terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART_TERMINAL' IN
TYPEOF (i_f.relate_shape_aspect)) AND
(i_f.relate_shape_aspect\shape_aspect.description =
'interface terminal')))) = 1);
END_ENTITY; -- component_interface_terminal
(*

```

Formal propositions:

WR1: The component_interface_terminal shall have a description of 'interconnect component interface terminal' or 'packaged connector component interface terminal'.

WR2: The component_interface_terminal shall be an aspect of the shape of a component_definition.

WR3: The `component_interface_terminal` shall be related to at most one `shape_aspect` as the 'instantiated feature'.

WR4: The `component_interface_terminal` shall relate to at most one `shape_aspect` to specify the 'design usage'.

WR5: If the name of the `component_interface_terminal` is 'interconnect component interface terminal', it shall be an aspect of the shape of an `interface_component_definition`.

WR6: If the name of the `component_interface_terminal` is 'interconnect component interface terminal', it shall be an aspect of the shape of a `product_definition` with a `frame_of_reference` name of 'physical occurrence'.

WR7: If the `component_interface_terminal` has a description of 'interconnect component interface terminal', the `component_interface_terminal` shall relate to exactly one `shape_aspect` that has a description of 'connection zone' and that is associated with a product that is categorized as an 'interconnect module' and does not have a description of 'assembly module component', 'bare die component', 'interconnect module component', 'laminated component', or 'packaged component'.

WR8: If the name of the `component_interface_terminal` is 'packaged connector component interface terminal', it shall lie on the boundary of the shape of the `component_definition`.

WR9: If the `component_interface_terminal` has a description of 'packaged connector component interface terminal', it shall be an aspect of the shape of a `packaged_component` that has a description of 'packaged connector component' and a `frame_of_reference` name of 'physical occurrence'.

WR10: If the `component_interface_terminal` has a description of 'packaged connector component interface terminal', the `component_interface_terminal` shall be related to exactly one `packaged_part_terminal` with a description of 'interface terminal' as the 'instantiated feature'.

5.2.3.1.69 `component_location`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `component_location` is a representation that implements the ARM concepts of:

- `Component_2d_edge_location` (see 4.2.106);
- `Component_2d_location` (see 4.2.108);
- `Component_2d_stacked_location` (see 4.2.109);
- `Component_3d_location` (see 4.2.111).

EXPRESS specification:

```

*)
ENTITY component_location
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'placement fixed') AND
    ((it\descriptive_representation_item.description = 'true') OR
    (it\descriptive_representation_item.description = 'false')))) = 1;

  WR2: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)))) = 1;

  WR3: (NOT
    (SELF.context_of_items\geometric_representation_context.
    coordinate_space_dimension
    = 2)) OR (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (it))) = 1);

  WR4: (NOT
    (SELF.context_of_items\geometric_representation_context.
    coordinate_space_dimension
    = 3)) OR (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D' IN TYPEOF (it))) = 1);

  WR5: SIZEOF (QUERY (it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'
    IN TYPEOF (it))
    )) = 1;

```

```

WR6: NOT (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (it))) = 1)
OR
(SIZEOF (QUERY (cto2d <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_TRANSFORMATION_OPERATOR_2D'
    IN TYPEOF (cto2d)) AND
    (SIZEOF (QUERY (mi <* USEDIN (cto2d,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM.MAPPING_TARGET') | (
    (SIZEOF (QUERY (cl <* USEDIN (mi,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') |(cl = SELF))) = 1)))) >= 1)
    )) = 1);

WR7: NOT (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D' IN TYPEOF (it))) = 1)
OR
(SIZEOF (QUERY (cto2d <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D'
    IN TYPEOF (cto2d)) AND
    (SIZEOF (QUERY (mi <* USEDIN (cto2d,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM.MAPPING_TARGET') | (
    (SIZEOF (QUERY (cl <* USEDIN (mi,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') |(cl = SELF))) = 1)))) = 1)
    )) = 1);

```

```

WR8:NOT((SELF.context_of_items\geometric_representation_context.coordinate_space_dimensionality
= 2) AND
  (SELF.context_of_items.context_type = 'component surface') AND
  (( SIZEOF (QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
(
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
(pdr.name = 'mounting surface assembly joint') AND
(pdr.definition.definition.relating_shape_aspect\shape_aspect.description =
'interconnect module component surface feature') AND
(SIZEOF(QUERY ( sar <* USEDIN (pdr.definition.definition.relating_shape_aspect,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.relating_shape_aspect\shape_aspect.description =
'interconnect module primary surface') OR
(sar.relating_shape_aspect\shape_aspect.description =
'interconnect module edge surface') OR
(sar.relating_shape_aspect\shape_aspect.description =
'interconnect module edge segment surface')))) = 1)
)
)) = 1)))
OR
(SIZEOF(QUERY (cto2d <* SELF.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (cto2d)) AND
(cto2d_determinant_test(cto2d,1.0,0.001 )))) = 1);

```



```

WR9: NOT((SELF.context_of_items\geometric_representation_context.coordinate_space_dimension
    = 2) AND
    (SELF.context_of_items.context_type = 'component surface') AND
    (( SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    (
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
    (pdr.name = 'mounting surface assembly joint') AND
    (pdr.definition.definition.relate_shape_aspect\shape_aspect.description =
    'interconnect module component surface feature') AND
    (SIZEOF(QUERY ( sar <* USEDIN (pdr.definition.definition.relate_shape_aspect,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar.relate_shape_aspect\shape_aspect.description =
    'interconnect module secondary surface')) = 1)
    ))) = 1)))
OR
(SIZEOF(QUERY( cto2d <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (cto2d)) AND
    (cto2d_determinant_test(cto2d,-1.0,0.001 )))) = 1);

WR10: (NOT(is_laminate_component_location(SELF))) OR
    (SIZEOF(QUERY( pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
    (pdr.definition.definition.relate_shape_aspect\shape_aspect.description =
    'interconnect module component surface feature')))) = 0);

WR11: (NOT(is_laminate_component_location(SELF) AND
    (SELF.context_of_items\geometric_representation_context.coordinate_space_dimension
    = 2)))
OR
    (SIZEOF(QUERY (cto2d <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (cto2d)) AND
    (cto2d_determinant_test(cto2d,1.0,0.001 )))) = 1);

WR12: SIZEOF(QUERY( pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT' IN TYPEOF (pdr.definition.definition)) AND
    (pdr.definition.definition\shape_aspect.description =
    'interconnect module component surface feature')))) = 0;

```

```

WR13: (NOT((SELF.context_of_items\geometric_representation_context.
coordinate_space_dimension = 2) AND
(SELF.context_of_items.context_type = 'component stacked'))
OR
(SIZEOF(QUERY( pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
(pdr.definition.definition.relatng_shape_aspect\shape_aspect.description =
'interconnect module component surface feature')))) = 0);

```

```

WR14: NOT((SELF.context_of_items\geometric_representation_context.coordinate_space_dimensions
    = 2)
    AND
    (SELF.context_of_items.context_type = 'component edge'))
OR
    ((NOT(is_laminate_component_location(SELF)))
    AND
    (( SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    (
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
    (pdr.name = 'reference terminal assembly joint') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (pdr.definition.definition.related_shape_aspect))
    AND
    (pdr.definition.definition.related_shape_aspect\shape_aspect.description IN
    ['packaged component join terminal',
    'package terminal occurrence']))
    AND
    (SIZEOF(QUERY ( sar <* USEDIN
    (pdr.definition.definition.related_shape_aspect,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'instantiated feature')
    AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRIMARY_REFERENCE_TERMINAL' IN TYPEOF (sar.relate_shape_aspect))
    )) = 1)
    AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (pdr.definition.definition.related_shape_aspect.of_shape.definition))
    AND
    (component_definition_located_by_component_location(SELF) =
    pdr.definition.definition.related_shape_aspect.of_shape.definition)
    ))) = 1))
    AND
    (( SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    (
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
    (pdr.name = 'mounting surface assembly joint') AND
    (pdr.definition.definition.relate_shape_aspect\shape_aspect.description =
    'interconnect module component surface feature') AND
    (SIZEOF(QUERY ( sar <* USEDIN (pdr.definition.definition.relate_shape_aspect,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |

```

```

(sar.relating_shape_aspect\shape_aspect.description =
'interconnect module edge surface') OR
(sar.relating_shape_aspect\shape_aspect.description =
'interconnect module edge segment surface')) = 1)
AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_MOUNTING_FEATURE' IN
    TYPEOF (pdr.definition.definition.related_shape_aspect))
AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION' IN
    TYPEOF (pdr.definition.definition.related_shape_aspect.of_shape.definition))
AND
(component_definition_located_by_component_location(SELF) =
    pdr.definition.definition.related_shape_aspect.of_shape.definition)
))) = 1))
);

```

```

WR15: NOT((SELF.context_of_items\geometric_representation_context.coordinate_space_dimensions
    = 2)
    AND
    (SELF.context_of_items.context_type = 'component stacked'))
OR
    ((NOT(is_laminate_component_location(SELF)))
    AND
    (( SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    (
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition))
    AND
    ('mounting joint' = pdr.name)
    AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_MOUNTING_FEATURE' IN
        TYPEOF (pdr.definition.definition.related_shape_aspect))
    AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
        TYPEOF (pdr.definition.definition.related_shape_aspect.of_shape.definition))
    AND
    (component_definition_located_by_component_location(SELF) ::=
        pdr.definition.definition.related_shape_aspect.of_shape.definition)
    AND
    (component_definition_located_by_component_location(SELF) :<:
        pdr.definition.definition.relating_shape_aspect.of_shape.definition)
    )
    )) = 1))
);

```

```

WR16: (NOT(is_interconnect_module_component_location(SELF))) OR
      (SIZEOF(QUERY( pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_JOINT' IN TYPEOF (pdr.definition.definition)) AND
        (pdr.definition.definition.relatng_shape_aspect\shape_aspect.description =
        'interconnect module component surface feature')))) = 0);

WR17: NOT((SELF.context_of_items\geometric_representation_context.coordinate_space_dimensions
      = 2)
      AND
      (SELF.context_of_items.context_type = 'component stacked'))
      OR
      (SIZEOF(QUERY (cto2d <* SELF.items |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF (cto2d)) AND
        ((cto2d_determinant_test(cto2d,1.0,0.001 )) OR
        (cto2d_determinant_test(cto2d,-1.0,0.001 )))
      )) = 1);

WR18 : SIZEOF (QUERY (it <* SELF\representation.items| NOT
      (SIZEOF(
      ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CARTESIAN_TRANSFORMATION_OPERATOR_2D',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_3D']
      * TYPEOF(it)) = 1 ))) = 0;

WR19: SIZEOF (QUERY (it <* SELF.items |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_3D'
      IN TYPEOF (it)) AND
      (it\representation_item.name = 'origin')
      )) = 0;

```

```

WR20: SIZEOF (QUERY (it <* SELF.items |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MAPPED_ITEM'
  IN TYPEOF (it))
  AND NOT
  (((it\representation_item.name = 'component assembly 2d position')
  OR
  (it\representation_item.name = 'component assembly 3d position'))
  AND
  (it.mapping_source.mapping_origin\representation_item.name = 'origin')
  AND
  (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'AXIS2_PLACEMENT_2D' IN TYPEOF (it.mapping_source.mapping_origin))
  OR
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'AXIS2_PLACEMENT_3D' IN TYPEOF (it.mapping_source.mapping_origin))))
  )) = 0;

WR21: SIZEOF(QUERY(pdr <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
  + 'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
  + 'NEXT_ASSEMBLY_USAGE_OCCURRENCE' IN
  TYPEOF(pdr.definition.definition))
  )) = 1;
END_ENTITY; -- component_location
(*

```

Formal propositions:

WR1: The component_location shall contain exactly one descriptive_representation_item with a name 'placement fixed' and the description of that descriptive_representation_item shall be either 'true' or 'false'.

WR2: The component_location shall contain exactly one descriptive_representation_item.

WR3: If the component_location has a geometric_representation_context with a coordinate_space_dimension value of 2, it shall contain exactly one cartesian_transformation_operator_2d.

WR4: If the component_location has a geometric_representation_context with a coordinate_space_dimension value of 3, it shall contain exactly one axis2_placement_3d.

WR5: The component_location shall contain exactly one mapped_item.

WR6: The component_location that contains a cto2d shall contain exactly one mapped_item that specifies the cto2d.

WR7: The component_location that contains an a2p3d shall contain exactly one mapped_item that specifies the a2p3d.

WR8: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and if the `component_location` references a primary surface, edge surface, or edge segment surface, the determinant of the `cto2d` shall be +1.0 within a delta of 0.001.

WR9: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and if the `component_location` references a secondary surface, the determinant of the `cto2d` shall be -1.0 within a delta of 0.001.

WR10: If the `component_location` is the location of the implementation of the ARM concept of `Laminate_component`, it shall not be associated with a `component_shape_aspect` where the description is 'interconnect module component surface feature'.

WR11: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and is the location of the implementation of the ARM concept of `Laminate_component`, the determinant of the `cto2d` shall be +1.0 within a delta of 0.001.

WR12: The `component_location` shall not directly reference any `component_shape_aspect` where the description is 'interconnect module component surface feature'.

WR13: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and a `context_type` of 'component stacked', it shall not reference a `component_shape_aspect` where the description is 'interconnect module component surface feature'.

WR14: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and a `context_type` of 'component edge', it shall be associated with exactly one `assembly_joint` where the `assembly_joint` shall specify with the `related_shape_aspect` a `component_terminal` that is defined by a `primary_reference_terminal`. The `component_location` shall be associated with a `component_shape_aspect` where the description is 'interconnect module component surface feature' that is defined by an edge or edge_segment of an interconnect module. The `component_terminal` specified by the `related_shape_aspect` of the `assembly_joint` shall be a `component_terminal` of the `component_definition` being located by this `component_location`.

WR15: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and a `context_type` of 'component stacked', it shall be associated with exactly one `assembly_joint` where the `assembly_joint` shall specify with the `related_shape_aspect` a `component_mounting_feature`. The `component_mounting_feature` referenced by the `assembly_joint` `related_shape_aspect` shall be a `component_mounting_feature` of the `component_definition` being located by this `component_location`. The `relating_shape_aspect` of the `assembly_joint` shall specify a `component_feature` of a `component_definition` that is not being located by this `component_location`.

NOTE The base component of the stack is not considered to be stacked. The `assembly_joint` referenced to support stacking are directed for stack traversal.

EXAMPLE 1 In this example, three bare die are vertically mounted on top of a substrate. The substrate is not considered to be part of the stack. The first bare die is not considered to be 'stacked'. The `component_location` references the substrate. The second bare die is considered to be 'stacked'. The `component_location` references an `assembly_joint`. The `related_shape_aspect` of the `assembly_joint` references the second bare die, and the `relating_shape_aspect` of the `assembly_joint` references the first bare die. The third bare die is considered to be 'stacked'. The `component_location` references a different `assembly_joint`. The `related_shape_aspect` of the `assembly_joint` references the third bare die, and the `relating_shape_aspect` of the `assembly_joint` references the second bare die.

WR16: If the `component_location` is the location of the implementation of the ARM concept of `Interconnect_module_component`, it shall not be associated with a `component_shape_aspect` where the description is 'interconnect module component surface feature'.

WR17: If the `component_location` has a `geometric_representation_context` with a `coordinate_space_dimension` value of 2 and a `context_type` of 'component stacked', the determinant of the `cto2d` shall be +1.0 or -1.0, in either case within a delta of 0.001.

WR18: The items in a `component_location` shall be a `mapped_item`, a `descriptive_representation_item`, a `cartesian_transformation_operator_2d`, or an `axis2_placement_3d`.

WR19: The `axis2_placement_3d` shall not have a name of 'origin'.

NOTE This does not prohibit placing the component at the origin. It prohibits an `axis2_placement_3d` that is in the role of 'origin' from serving as the location.

WR20: The `representation_item` used as a `mapping_source` shall be either an `axis2_placement_2d` or shall be an `axis2_placement_3d` and it shall have a name of 'origin'.

WR21: The `component_location` shall be used by exactly one `next_assembly_usage_occurrence`.

5.2.3.1.70 component_mating_constraint_condition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `component_mating_constraint_condition` is a `shape_aspect` that implements the ARM concept of `Component_mating_constraint_condition` (see 4.2.127).

EXPRESS specification:

```
*)
ENTITY component_mating_constraint_condition
  SUBTYPE OF (shape_aspect);
UNIQUE
  UR1: SELF\shape_aspect.name, SELF\shape_aspect.of_shape;
END_ENTITY; -- component_mating_constraint_condition
(*
```

Formal propositions:

UR1: The name shall be unique within the context of the `product_definition_shape` referenced as `of_shape` inherited from the `shape_aspect` supertype.

WR1: The `component_mating_constraint_condition` shall relate to at least one and no more than three `assembly_joint` to specify the 'constraining_joint'.

WR2: The `component_mating_constraint_condition` shall be related to at least one and no more than five `representation_items` to describe the 'degree of freedom' constrained.

WR3: The component_mating_constraint_condition shall not be any other type of shape_aspect.

5.2.3.1.71 component_mounting_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A component_mounting_feature is a shape_aspect. A component_mounting_feature implements the ARM concept of Component_mounting_feature (see 4.2.129)

EXPRESS specification:

```

*)
ENTITY component_mounting_feature
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);

  WR2: (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTER_STRATUM_FEATURE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROBE_ACCESS_AREA',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND'] *
        TYPEOF (SELF)) = 0);

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) |
        sr_pdr.used_representation\representation.name =
        'planar projected shape'))
        <= 1))) = 0;

```

```

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)) |
    sr_pdr.used_representation\representation.name = '3d bound volume shape'))
    <= 1))) = 0;

WR5: (NOT (SELF\shape_aspect.description IN ['component feature'])) OR
    (SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'design usage')) <= 1);

WR6: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature')) <= 1;
END_ENTITY; -- component_mounting_feature
(*

```

Formal propositions:

- WR1:** The component_mounting_feature shall be an aspect of the shape of a component_definition.
- WR2:** The component_mounting_feature shall not be any of the following: inter_stratum_feature, probe_access_area, land.
- WR3:** The component_mounting_feature shall have at most one shape_representation with a name of 'planar projected shape'.
- WR4:** The component_mounting_feature shall have at most one shape_representation with a name of '3d bound volume shape'.
- WR5:** If the component_mounting_feature has a description of 'component feature', it shall relate to at most one shape_aspect to specify the 'design usage'.
- WR6:** The component_mounting_feature shall be related to at most one shape_aspect as an 'instantiated feature'.

5.2.3.1.72 component_shape_aspect

A component_shape_aspect is a shape_aspect. A component_shape_aspect implements the ARM concepts of:

- Conductive_interconnect_element (see 4.2.171);
- Conductive_interconnect_element_with_pre_defined_transitions (see 4.2.174);
- Conductive_interconnect_element_with_user_defined_single_transition (see 4.2.175);
- Join_two_physical_connectivity_definition_supporting_printed_component (see 4.2.481);
- Laminate_component (see 4.2.483);
- Laminate_text_component (see 4.2.488);
- Laminate_text_string_component (see 4.2.489);
- Primary_stratum_indicator_symbol (see 4.2.719);
- Special_symbol_laminate_component (see 4.2.834);
- Stratum_feature_template_component (see 4.2.862);
- Thermal_component (see 4.2.900).

EXPRESS specification:

```

*)
ENTITY component_shape_aspect
  SUPERTYPE OF (ONEOF (land, connected_area_component, inter_stratum_feature)
                 ANDOR (thermal_component_shape_aspect))
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);

```

```

WR2: (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FIDUCIAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONNECTED_AREA_COMPONENT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_GROUP_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'THERMAL_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MATERIAL_REMOVAL_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTEGRAL_SHIELD',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROBE_ACCESS_AREA',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND'] *
  TYPEOF (SELF)) >= 1) OR
(SELF\shape_aspect.description IN [
'part template occurrence',
'laminate component',
'laminate text component',
'laminate text string component',
'conductive interconnect element with pre defined transitions',
'join 2 physical connectivity definition supporting',
'conductive interconnect element with user defined single transition',
'special symbol laminate component',
'primary stratum indicator symbol',
'stratum feature template component']);

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) |
sr_pdr.used_representation\representation.name = 'planar projected shape'))
<= 1))) = 0;

```

```

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)) |
    sr_pdr.used_representation\representation.name = '3d bound volume shape'))
    <= 1))) = 0;

WR5: NOT(EXISTS(SELF\shape_aspect.description)) OR
    (NOT (SELF\shape_aspect.description IN ['laminate text component',
    'laminate text string component',
    'special symbol laminate component',
    'primary stratum indicator symbol',
    'stratum feature template component'])) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition)) AND
    (SELF.of_shape.definition\product_definition.description =
    'laminate component')
    ));

WR6: NOT(EXISTS(SELF\shape_aspect.description)) OR
    (NOT (SELF\shape_aspect.description =
    'laminate text component') OR
    (SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated template') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEXT_TEMPLATE_DEFINITION' IN
    TYPEOF (it.relateing_shape_aspect)))) = 1));

WR7: NOT(EXISTS(SELF\shape_aspect.description)) OR
    (NOT (SELF\shape_aspect.description IN
    ['stratum feature template component'])) OR
    (SIZEOF (QUERY (sfi <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'stratum feature implementation') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' IN
    TYPEOF (sfi.relateing_shape_aspect)))) = 1));

```

```

WR8: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description IN
        ['conductive interconnect element with pre defined transitions',
        'join 2 physical connectivity definition supporting',
        'conductive interconnect element with ' +
        'user defined single transition'])) OR
      (SIZEOF (QUERY (cc <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'composed conductor') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE' IN
        TYPEOF (cc.related_shape_aspect)))) = 1));
(*)

WR9: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description IN
        ['conductive interconnect element with pre defined transitions',
        'join 2 physical connectivity definition supporting',
        'conductive interconnect element with user ' +
        'defined single transition'])) OR
      (SIZEOF (QUERY (at <* associated_terminals(SELF)|
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL' IN
        TYPEOF (at)) AND
        (at\shape_aspect.description =
        'conductive interconnect element terminal')))) >= 2));
*)

WR10: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description IN
        ['special symbol laminate component',
        'primary stratum indicator symbol'])) OR
      (SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'instantiated template') |
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION' IN
        TYPEOF (it.relatating_shape_aspect)) AND
        (it.related_shape_aspect\shape_aspect.description =
        'special symbol part template')))) = 1));

```

```

WR11: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description =
        'stratum feature template component') OR
        (SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'instantiated template') |
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION' IN
        TYPEOF (it.relateing_shape_aspect)) AND
        (it.relateing_shape_aspect\shape_aspect.description =
        'stratum feature template')) OR
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_PHYSICAL_TEMPLATE' IN
        TYPEOF (it.relateing_shape_aspect)))) <= 1));

WR12: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description IN
        ['laminate text component'])) OR
        (SIZEOF (QUERY (sfi <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'stratum feature implementation') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE' IN
        TYPEOF (sfi.relateing_shape_aspect)))) >= 1));

```

```

WR13: NOT ((SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FIDUCIAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONNECTED_AREA_COMPONENT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_GROUP_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'THERMAL_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MATERIAL_REMOVAL_COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTEGRAL_SHIELD',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROBE_ACCESS_AREA',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND'] *
TYPEOF (SELF)) >= 1) OR
(SELF\shape_aspect.description IN [
'laminate component',
'laminate text component',
'laminate text string component',
'conductive interconnect element with pre defined transitions',
'join 2 physical connectivity definition supporting',
'conductive interconnect element with user defined single transition',
'special symbol laminate component',
'primary stratum indicator symbol',
'stratum feature template component'])) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION' IN
TYPEOF (SELF.of_shape.definition)) AND
(SELF.of_shape.definition\
product_definition.frame_of_reference\
application_context_element.name = 'layout occurrence'));

WR14: SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template')) <= 1;

```



```

WR15: NOT(EXISTS(SELF\shape_aspect.description)) OR
      (NOT (SELF\shape_aspect.description =
        'join 2 physical connectivity definition supporting') OR
      (SIZEOF ( QUERY (propd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF (QUERY (ri <* QUERY (propdr <* USEDIN (propd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      propdr\property_definition_relationship.name = 'requirement implementation') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP' IN
      TYPEOF (ri.relying_property_definition.definition)) AND
      (ri.relying_property_definition.definition\shape_aspect_relationship.name =
        'ordered physical connectivity definition') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_CONNECTIVITY_DEFINITION'
      IN TYPEOF (ri.relying_property_definition.definition.
        related_shape_aspect)) AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_CONNECTIVITY_DEFINITION' IN TYPEOF
      (ri.relying_property_definition.definition.
        relating_shape_aspect)))) = 1))) = 1));
END_ENTITY; -- component_shape_aspect
( *

```

Formal propositions:

WR1: The component_shape_aspect shall be an aspect of the shape of a component_definition.

WR2: The component_shape_aspect shall be an inter_stratum_feature, probe_access_area, or land, or fiducial, or assembly_group_component_shape_aspect, or material_removal_component_shape_aspect, or integral_shield, or thermal_component_shape_aspect or connected_area_component, or it shall have a description of 'laminate component', 'laminate text component', 'laminate text string component', 'part template occurrence', 'conductive interconnect element with pre defined transitions', 'join 2 physical connectivity definition supporting', 'conductive interconnect element with user defined single transition', 'special symbol laminate component', 'primary stratum indicator symbol', or 'stratum feature template component'.

WR3: The component_shape_aspect shall have at most one shape_representation with a name of 'planar projected shape'.

WR4: The component_shape_aspect shall have at most one shape_representation with a name of '3d bound volume shape'.

WR5: If the component_shape_aspect has a description of 'laminate text component', 'laminate text string component', 'special symbol laminate component', 'primary stratum indicator symbol', or 'stratum feature template component', it shall be an aspect of the shape of a component_definition that is the definition of a product that is categorized as an 'interconnect module'.

WR6: If the `component_shape_aspect` has a description of 'laminar text component', the `component_shape_aspect` shall be related to exactly one `part_text_template_definition`.

WR7: If the `component_shape_aspect` has a description of 'stratum feature template component', the `component_shape_aspect` shall be related to exactly one `stratum_feature` as the 'stratum feature implementation'.

WR8: If the `component_shape_aspect` has a description of either 'conductive interconnect element with pre defined transitions', 'join 2 physical connectivity definition supporting' or 'conductive interconnect element with user defined single transition', the `component_shape_aspect` shall relate to exactly one `stratum_feature` to specify the 'composed conductor'.

WR9: If the `component_shape_aspect` has a description of either 'conductive interconnect element with pre defined transitions', 'join 2 physical connectivity definition supporting' or 'conductive interconnect element with user defined single transition', the `component_shape_aspect` shall relate to at least two `component_terminal` that have a description of 'conductive interconnect element terminal' to specify the 'associated component'.

WR10: If the `component_shape_aspect` has a description of either 'special symbol laminar component' or 'primary stratum indicator symbol', the `component_shape_aspect` shall be related to exactly one `part_template_definition` with a description of 'special symbol part template' as an 'instantiated template'.

WR11: If the `component_shape_aspect` has a description of 'stratum feature template component', the `component_shape_aspect` shall be related, as an 'instantiated template', to not more than one `part_template_definition` that has a description of 'stratum feature template' or to not more than one `land_physical_template`.

WR12: If the `component_shape_aspect` has a description of 'laminar text component', the `component_shape_aspect` shall be related to at least one `stratum_feature` as the 'stratum feature implementation'.

WR13: If the `component_shape_aspect` is an `inter_stratum_feature`, `probe_access_area`, or `land`, or `fiducial`, or `assembly_group_component_shape_aspect`, or `material_removal_component_shape_aspect`, or `integral_shield`, or `thermal_component_shape_aspect`, or `connected_area_component`, or if it has a description of 'laminar component', 'laminar text component', 'laminar text string component', 'conductive interconnect element with pre defined transitions', 'join 2 physical connectivity definition supporting', 'conductive interconnect element with user defined single transition', 'special symbol laminar component', 'primary stratum indicator symbol', or 'stratum feature template component', it shall be a `shape_aspect` for a `component_definition` with a `frame_of_reference` of 'layout occurrence'.

WR14: The `component_shape_aspect` shall be related to at most one `shape_aspect` as an 'instantiated template'.

WR15: If the `component_shape_aspect` has a description of 'join 2 physical connectivity definition supporting', the `component_shape_aspect` shall have exactly one `property_definition` that is related, as the 'requirement implementation', to exactly one `property_definition` that is the property of a `shape_aspect_relationship` that relates one `physical_connectivity_definition` to another `physical_connectivity_definition` as an 'ordered physical connectivity definition'.

Informal propositions:

IP1: If the `component_shape_aspect` has a description of 'conductive interconnect element with pre defined transitions', there shall be at least one `shape_aspect_relationship` with name of 'connected points' that relates the `component_shape_aspect` (as `relating_shape_aspect`) to one `layer_connection_point_link` (as `related_shape_aspect`).

IP2: If the `component_shape_aspect` has a description of 'conductive interconnect element with pre defined transitions' or 'conductive interconnect element with user defined single transition', each `shape_aspect_relationship` that uses the `component_shape_aspect` as its `relating_shape_aspect` and has a name of 'connected points' shall have a unique description.

IP3: The aggregation of design views referenced by the path in the ARM of `stratum_feature_implementation.resident_stratum.assembly.assembly` shall be the same aggregation of design views as that reference by the path in the ARM of `assembly.assembly` when the `component_shape_aspect` is an implementation of the ARM concept of `Stratum_feature_template_component`

5.2.3.1.73 component_terminal

A `component_terminal` is a `shape_aspect` that implements the ARM concepts of:

- `Assembly_module_component_terminal` (see 4.2.45);
- `Assembly_module_macro_component_join_terminal` (see 4.2.49);
- `Bare_die_component_terminal` (see 4.2.62);
- `Component_terminal` (see 4.2.137);
- `Component_termination_passage_join_terminal` (see 4.2.149);
- `Conductive_interconnect_element_terminal` (see 4.2.172);
- `Interconnect_component_join_terminal` (see 4.2.439);
- `Interconnect_module_component_terminal` (see 4.2.445);
- `Interconnect_module_macro_component_join_terminal` (see 4.2.456);
- `Land_join_terminal` (see 4.2.492);
- `Minimally_defined_component_terminal` (see 4.2.546);
- `Non_functional_land_join_terminal` (see 4.2.564);
- `Package_terminal_occurrence` (see 4.2.583);

- Packaged_component_join_terminal (see 4.2.586);
- Printed_component_join_terminal (see 4.2.722);
- Via_terminal (see 4.2.954).

EXPRESS specification:

```

*)
ENTITY component_terminal
  SUPERTYPE OF (ONEOF(assembly_module_macro_component_join_terminal,
    interconnect_module_macro_component_join_terminal))
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: (SELF\shape_aspect.description IN
    ['assembly module component terminal',
    'bare die component terminal',
    'component termination passage join terminal',
    'conductive interconnect element terminal',
    'interconnect component join terminal',
    'interconnect module component terminal', 'land join terminal',
    'minimally defined component terminal',
    'non functional land join terminal',
    'packaged component join terminal',
    'printed component join terminal', 'package terminal occurrence',
    'via terminal']) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_MODULE_MACRO_COMPONENT_JOIN_TERMINAL' IN
    TYPEOF (SELF)) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERCONNECT_MODULE_MACRO_COMPONENT_JOIN_TERMINAL' IN
    TYPEOF (SELF));

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature')) <= 1;

```

```

WR4: SIZEOF (QUERY (at <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'associated terminals') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_CONNECTIVITY_DEFINITION' IN
TYPEOF (at.relateing_shape_aspect))) <= 1;

WR5: SIZEOF (QUERY (cr <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'connectivity requirement') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_NETWORK' IN
TYPEOF (cr.relateing_shape_aspect))) <= 1;

WR6: SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'design usage')) <= 1;

WR7: (NOT (SELF\shape_aspect.description IN
['assembly module component terminal',
'bare die component terminal', 'package terminal occurrence',
'packaged component join terminal',
'printed component join terminal'])) OR
(SELF.product_definitional);

WR8: (NOT (SELF\shape_aspect.description =
'assembly module component terminal')) OR
((SELF.of_shape.definition.frame_of_reference.name =
'physical occurrence') AND
(EXISTS(SELF.of_shape.definition\product_definition.name)) AND
(NOT(EXISTS(SELF.of_shape.definition\product_definition.name)) OR
(SELF.of_shape.definition\product_definition.name =
'assembly module'))
);

WR9: (NOT (SELF\shape_aspect.description =
'assembly module component terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_TERMINAL' IN
TYPEOF (i_f.relateing_shape_aspect)))) = 1);

```

```

WR10: (NOT (SELF\shape_aspect.description =
'bare die component terminal')) OR
(SELF.of_shape.definition\product_definition.description =
'bare die component');

```

```

WR11: (NOT (SELF\shape_aspect.description =
'bare die component terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'BARE_DIE_TERMINAL' IN
TYPEOF (i_f.relateing_shape_aspect)))) = 1);
(*)

```

```

WR12: (NOT (SELF\shape_aspect.description =
'component termination passage join terminal')) OR
(
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE' IN
TYPEOF (associated_component(SELF))) AND
(associated_component(SELF).description =
'component termination passage'));
*)

```

```

WR13: (NOT (SELF\shape_aspect.description =
'component termination passage join terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
i_f.relateing_shape_aspect\shape_aspect.description =
'component termination passage template join terminal')) = 1);
(*)

```

```

WR14: (NOT (SELF\shape_aspect.description =
'conductive interconnect element terminal')) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT' IN
TYPEOF (associated_component(SELF)) AND
((associated_component(SELF).description =
'conductive interconnect element with pre defined transitions') OR
(associated_component(SELF).description =
'conductive interconnect element with user defined single transition')
));
*)

```

```

WR15: (NOT (SELF\shape_aspect.description IN
['conductive interconnect element terminal',
'interconnect component join terminal',
'printed component join terminal'])) OR
(SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar.related_shape_aspect\shape_aspect.description =
'connection zone')) <= 1);

WR16: (NOT (SELF\shape_aspect.description =
'interconnect module component terminal')) OR
((SELF.of_shape.definition\product_definition.frame_of_reference.name =
'physical occurrence') AND
(EXISTS(SELF.of_shape.definition\product_definition.name)) AND
(NOT(EXISTS(SELF.of_shape.definition\product_definition.name)) OR
(SELF.of_shape.definition\product_definition.name =
'interconnect module')))
);

WR17: (NOT (SELF\shape_aspect.description =
'interconnect module component terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_TERMINAL' IN
TYPEOF (i_f.relying_shape_aspect)
)) = 1);

WR18: (NOT (SELF\shape_aspect.description IN
['land join terminal', 'non functional land join terminal'])) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_TEMPLATE_TERMINAL' IN
TYPEOF (i_f.relying_shape_aspect)) AND
(TRUE)))) = 1);
( *

WR19: (NOT (SELF\shape_aspect.description = 'land join terminal')) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND' IN
TYPEOF (associated_component(SELF)));
*)

```

```

WR20: (NOT (SELF\shape_aspect.description =
'minimally defined terminal')) OR
(SIZEOF (QUERY (pdr<* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'part terminal external reference')) = 1);
(*)

WR21: (NOT (SELF\shape_aspect.description =
'non functional land join terminal')) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND' IN
TYPEOF (associated_component(SELF))) AND
(associated_component(SELF).description IN
['non functional land',
'via dependent non functional land',
'via and contact size dependent non functional land',
'component termination passage dependent non functional land',
'contact size dependent non functional land',
'component termination passage and contact size ' +
'dependent non functional land',
'unsupported passage dependent non functional land'])))) = 1);
*)

WR22: (NOT (SELF\shape_aspect.description =
'package terminal occurrence')) OR
(SIZEOF (QUERY (ud <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'usage definition') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGE_TERMINAL' IN
TYPEOF (ud.relate_shape_aspect)))) = 1);

WR23: (NOT (SELF\shape_aspect.description IN
['package terminal occurrence',
'packaged component join terminal'])) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_COMPONENT' IN
TYPEOF (SELF.of_shape.definition)) AND
(SELF.of_shape.definition\product_definition.
frame_of_reference.name =
'physical occurrence'));

```



```

WR24: (NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART_TERMINAL' IN
TYPEOF (i_f.relating_shape_aspect)) AND
(i_f.relating_shape_aspect\shape_aspect.description =
'join terminal')))) = 1);

WR25: (NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2)) <= 2)) <= 1)) <= 1));

WR26: (NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (lmwu <* QUERY (it <* pdr.used_representation.items |
SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) |
lmwu\representation_item.name = 'maximum wire length')) <= 1)) <= 1)) <= 1);

```

```

WR27: (NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (lmwu <* QUERY (it <* pdr.used_representation.items |
SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) |
lmwu\representation_item.name = 'minimum wire length')) <= 1)) <= 1)) <= 1));

WR28: (NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR
((SELF.of_shape.definition\product_definition.
frame_of_reference.name =
'layout occurrence') AND
(SELF.of_shape.definition\product_definition.
name = 'interconnect module')
);

WR29: (NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR
(SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated feature') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL' IN
TYPEOF (i_f.relateing_shape_aspect)) AND
(i_f.relateing_shape_aspect\shape_aspect.description =
'join terminal')))) = 1);

WR30: (NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR
(SIZEOF (QUERY (i <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
pdr\shape_aspect_relationship.name = 'implementation') |
((SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT'] *
TYPEOF (i.relateing_shape_aspect)) >= 1) AND
(i.relateing_shape_aspect\shape_aspect.description =
'stratum feature template component'))
)) = 1);

```

(*

```

WR31: (NOT (SELF\shape_aspect.description = 'via terminal')) OR
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PLATED_PASSAGE' IN
       TYPEOF (associated_component(SELF))) AND
       (associated_component(SELF).description IN
        ['buried via', 'interfacial connection',
         'bonded conductive base blind via',
         'non conductive base blind via',
         'plated conductive base blind via']))) = 1);

```

*)

```

WR32: (NOT (SELF\shape_aspect.description = 'via terminal')) OR
      (SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
       'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
       sar\shape_aspect_relationship.name = 'instantiated feature') |
       i_f.relate_shape_aspect\shape_aspect.description =
       'via template terminal')) = 1);

```

```

WR33: (NOT (SELF\shape_aspect.description =
       'interconnect component join terminal')) OR
      ((SELF.of_shape.definition\product_definition.
       frame_of_reference.name =
       'physical occurrence') AND
       NOT(SELF.of_shape.definition\product_definition.description
        IN ['assembly module component', 'bare die component',
         'interconnect module component', 'lamine component',
         'packaged component']));

```

```

WR34: SIZEOF(QUERY(pd <* USEDIN(SELF,
       'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       SIZEOF(QUERY(dri <* pdr.used_representation.items |
       (dri\representation_item.name = 'global swappable') AND
       (dri\descriptive_representation_item.description IN ['true', 'false']))) = 1)) = 1)

```

```

WR35: SIZEOF(QUERY(pd <* USEDIN(SELF,
       'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       SIZEOF(QUERY(dri <* pdr.used_representation.items |
       (dri\representation_item.name = 'local swappable') AND
       (dri\descriptive_representation_item.description IN ['true', 'false']))) = 1)) = 1)

```

```

WR36: SIZEOF(QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF(QUERY(dri <* pdr.used_representation.items | dri\representation_item.name =
    'swap code'
    )) = 1)) = 1)) <= 1;
END_ENTITY; -- component_terminal
(*

```

Formal propositions:

WR1: The component_terminal shall have a description of 'assembly module component terminal', 'bare die component terminal', 'component termination passage join terminal', 'conductive interconnect element terminal', 'interconnect component join terminal', 'interconnect module component terminal', 'land join terminal', 'minimally defined component terminal', 'non functional land join terminal', 'packaged component join terminal', 'printed component join terminal', 'package terminal occurrence', or 'via terminal' or shall be an assembly_module_macro_component_join_terminal or shall be an interconnect_module_macro_component_join_terminal.

WR2: The component_terminal shall be an aspect of the shape of a component_definition.

WR3: The component_terminal shall be related to at most one shape_aspect as the 'instantiated feature'.

WR4: The component_terminal shall be related to at most one physical_connectivity_definition as 'associated terminals'.

WR5: The component_terminal shall be related to at most one physical_network as the 'connectivity requirement'.

WR6: The component_terminal shall relate to at most one shape_aspect to specify the 'design usage'.

WR7: If the component_terminal has a description of 'assembly module component terminal', 'bare die component terminal', 'package terminal occurrence', 'packaged component join terminal', or 'printed component join terminal', it shall lie on the boundary of the shape.

WR8: If the component_terminal has a description of 'assembly module component terminal', it shall be an aspect of the shape of a component_definition that has a frame_of_reference name of 'physical occurrence' or it shall be an assembly_module_macro_component_join_terminal and it shall be an aspect of the shape of a component_definition that is associated with a product that is a 'assembly module'.

WR9: If the component_terminal has a description of 'assembly module component terminal', the component_terminal shall be related to exactly one assembly_module_terminal as an 'instantiated feature'.

WR10: If the component_terminal has a description of 'bare die component terminal', it shall be an aspect of the shape of a product_definition that has a description of 'bare die component'.

WR11: If the component_terminal has a description of 'bare die component terminal', the component_terminal shall be related to exactly one bare_die_terminal as the 'instantiated feature'.

WR12: If the `component_terminal` has a description of 'component termination passage join terminal', the `component_terminal` shall be related to exactly one `plated_passage` with a description of 'component termination passage' as the 'associated component'.

WR13: If the `component_terminal` has a description of 'component termination passage join terminal', the `component_terminal` shall be related to exactly one `shape_aspect` with a description of 'component termination passage template join terminal' as the 'instantiated feature'.

WR14: If the `component_terminal` has a description of 'conductive interconnect element terminal', the `component_terminal` shall be related to exactly one `component_shape_aspect` with a description of either 'conductive interconnect element with pre defined transitions' or 'conductive interconnect element with user defined single transition' as the 'associated component'.

WR15: If the `component_terminal` has a description of 'conductive interconnect element terminal', 'interconnect component join terminal' or 'printed component join terminal', the `component_terminal` shall relate to zero or one `shape_aspect` that has a name of 'connection zone'.

WR16: If the `component_terminal` has a description of 'interconnect module component terminal', it shall be an aspect of the shape of a `component_definition` that has a `frame_of_reference` name of 'physical occurrence' or it shall be an `interconnect_module_macro_component_join_terminal` and it shall be an aspect of the shape of a `component_definition` that is associated with a product that is a 'interconnect module'.

WR17: If the `component_terminal` has a description of 'interconnect module component terminal', the `component_terminal` shall be related to exactly one `interconnect_module_terminal` as the 'instantiated feature' where the `interconnect_module_terminal` is on a `product_definition` of a product that is an 'interconnect module'.

WR18: If the `component_terminal` has a description of either 'land join terminal' or 'non functional land join terminal', the `component_terminal` shall be related to exactly one `land_template_terminal` as the 'instantiated feature'.

WR19: If the `component_terminal` has a description of 'land join terminal', the `component_terminal` shall be related to exactly one `land` as the 'associated component'.

WR20: If the `component_terminal` has a description of 'minimally defined component terminal' the `component_terminal` shall be the definition of exactly one `property_definition_representation` with a `used_representation.name` of 'part terminal external reference'.

WR21: If the `component_terminal` has a description of 'non functional land join terminal', the `component_terminal` shall be related to exactly one `land` that has a description of 'non functional land', 'via dependent non functional land', 'via and contact size dependent non functional land', 'component termination passage dependent non functional land', 'contact size dependent non functional land', 'component termination passage and contact size dependent non functional land', or 'unsupported passage dependent non functional land' as the 'associated component'.

WR22: If the `component_terminal` has a description of 'package terminal occurrence', the `component_terminal` shall be related to exactly one `package_terminal` as the 'usage definition'.

WR23: If the `component_terminal` has a description of 'package terminal occurrence', or 'packaged component join terminal' it shall be an aspect of the shape of a `packaged_component` that has a `frame_of_reference` name of 'physical occurrence'.

WR24: If the `component_terminal` has a description of 'packaged component join terminal', the `component_terminal` shall be related to exactly one `packaged_part_terminal` with a description of 'join terminal' as the 'instantiated feature'.

WR25: If the `component_terminal` has a description of 'packaged component join terminal', it shall have at most one `property_definition` that has at most one representation that contains at most two `length_measure_with_units`.

WR26: If the `component_terminal` has a description of 'packaged component join terminal', it shall have at most one `property_definition` that has at most one representation that contains at most one `length_measure_with_unit` with a name of 'maximum wire length'.

WR27: If the `component_terminal` has a description of 'packaged component join terminal', it shall have at most one `property_definition` that has at most one representation that contains at most one `length_measure_with_unit` with a name of 'minimum wire length'.

WR28: If the `component_terminal` has a description of 'printed component join terminal', it shall be an aspect of the shape of a `component_definition` that has a `frame_of_reference` name of 'layout occurrence' and that is associated with a product that is an 'interconnect module'.

WR29: If the `component_terminal` has a description of 'printed component join terminal', the `component_terminal` shall be related to exactly one `printed_part_template_terminal` with a description of 'join terminal' as the 'instantiated feature'.

WR30: If the `component_terminal` has a description of 'printed component join terminal', the `component_terminal` shall be related, as an 'implementation', to exactly one `component_shape_aspect` with a description of `stratum_feature_template_component`.

WR31: If the `component_terminal` has a description of 'via terminal', the `component_terminal` shall be related to exactly one `plated_passage` that has a description of 'buried via', 'interfacial connection', 'bonded conductive base blind via', 'non conductive base blind via', or 'plated conductive base blind via' as the 'associated component'.

WR32: If the `component_terminal` has a description of 'via terminal', the `component_terminal` shall be related to exactly one `shape_aspect` with a description of 'via template terminal' as the 'instantiated feature'.

WR33: If the `component_terminal` has a description of 'interconnect component join terminal', it shall be an aspect of the shape of a `component_definition` that has a `frame_of_reference` name of 'physical occurrence' and that is associated with a product that does not have a description of 'assembly module component', 'bare die component', 'interconnect module component', 'laminated component', or 'packaged component'.

WR34: The `component_terminal` shall be the definition for at most one `property_definition` that is the definition of a `property_definition_representation` that has a used representation that has an item that is a `descriptive_representation_item` with a name of 'global swappable' and a description of 'true' or 'false'.

WR35: The `component_terminal` shall be the definition for at most one `property_definition` that is the definition of a `property_definition_representation` that has a used representation that has an item that is a `descriptive_representation_item` with a name of 'local swappable' and a description of 'true' or 'false'.

WR36: The `component_terminal` shall be the definition for at most one `property_definition` that is the definition of a `property_definition_representation` that has a used representation that has an item that is a `descriptive_representation_item` with a name of 'swap code'.

Informal propositions:

IP1: If the `component_terminal` has a description of either 'land join terminal' or 'non functional land join terminal', the `component_terminal` shall be associated with the stratum referenced by the `stratum_surface`.

5.2.3.1.74 `composite_array_shape_aspect`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `composite_array_shape_aspect` is a `composite_shape_aspect` that implements the ARM concept of `Part_array_feature` (see 4.2.604) .

EXPRESS specification:

```
*)
ENTITY composite_array_shape_aspect
  SUPERTYPE OF (ONEOF( linear_composite_array_shape_aspect,
                        rectangular_composite_array_shape_aspect))
  SUBTYPE OF (composite_shape_aspect);
END_ENTITY; -- composite_array_shape_aspect
(*
```

5.2.3.1.75 `composite_array_shape_aspect_link`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `composite_array_shape_aspect_link` is a `shape_aspect` and a `shape_aspect_relationship` that implements the ARM concept of `Part_feature_placement_link` (see 4.2.612)

EXPRESS specification:

```
*)
ENTITY composite_array_shape_aspect_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE
```

```

WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
      SELF\shape_aspect_relationship.relate_shape_aspect;

WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT' IN
      TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT' IN
      TYPEOF (SELF\shape_aspect_relationship.relate_shape_aspect);

WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
      SELF\shape_aspect_relationship ||
      SELF\composite_array_shape_aspect_link)) = 0;

WR5: SELF\shape_aspect.name = '';

WR6: SELF\shape_aspect_relationship.name = '';

WR7: SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')) = 1;
END_ENTITY; -- composite_array_shape_aspect_link
(*

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The composite_array_shape_aspect_link shall relate to a shape_aspect in the related_shape_aspect.

WR3: The composite_array_shape_aspect_link shall relate to a shape_aspect in the relating_shape_aspect.

WR4: The types of composite_array_shape_aspect_link shall be only composite_array_shape_aspect_link, shape_aspect, and shape_aspect_relationship.

WR5: The name attribute of the shape_aspect supertype shall be an empty string.

WR6: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

WR7: Each composite_array_shape_aspect_link shall be referenced exactly once by a shape_aspect_relationship related_shape_aspect.

5.2.3.1.76 composite_group_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `composite_group_shape_aspect` is a `composite_shape_aspect` that implements the ARM concept of `Composite_group_shape_element` (see 4.2.159).

EXPRESS specification:

```
*)
ENTITY composite_group_shape_aspect
  SUBTYPE OF (composite_shape_aspect);
END_ENTITY; -- composite_group_shape_aspect
(*
```

5.2.3.1.77 `composite_signal_property_relationship`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `composite_signal_property_relationship` is a `property_definition_relationship` that implements the ARM concept of `Composite_signal_property_relationship` (see 4.2.164)

EXPRESS specification:

```
*)
ENTITY composite_signal_property_relationship
  SUBTYPE OF (property_definition_relationship);
WHERE

  WR1: SELF\property_definition_relationship.related_property_definition.definition :<>:
        SELF\property_definition_relationship.relying_property_definition.definition;

  WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\property_definition_relationship ||
        SELF\composite_signal_property_relationship || SELF\property_definition)) = 0;

  WR3: SELF\property_definition_relationship.related_property_definition.name =
        'signal property';

  WR4: SELF\property_definition_relationship.relying_property_definition.name =
        'composite signal property';

  WR5: SELF\property_definition.name = '';

  WR6: SELF\property_definition.description = '';

  WR7: SELF\property_definition_relationship.name = '';
```

```

WR8: SELF\property_definition_relationship.description = '';

WR9: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CHARACTERIZED_OBJECT' IN TYPEOF(SELF\property_definition.definition)) AND
      (SELF\property_definition.definition.description = 'aggregate operation');

WR10: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNAL_DEFINITION' IN TYPEOF(SELF\property_definition.definition);
END_ENTITY; -- composite_signal_property_relationship
(*)

```

Formal propositions:

WR1: The composite_signal_property_relationship shall not relate to the same property_definition definition through the related and relating attributes.

WR2: The composite_signal_property_relationship shall not be any other subtype of property_definition_relationship or property_definition.

WR3: The composite_signal_property_relationship shall reference only a 'signal property' through the related attribute.

WR4: The composite_signal_property_relationship shall reference only a 'composite signal property' through the relating attribute.

WR5: The composite_signal_property_relationship name inherited from property_definition shall be blank.

WR6: The composite_signal_property_relationship description inherited from property_definition shall be blank.

WR7: The composite_signal_property_relationship name inherited from property_definition_relationship shall be blank.

WR8: The composite_signal_property_relationship description inherited from property_definition_relationship shall be blank.

WR9: The composite_signal_property_relationship shall reference exactly one characterized_object that has a description of 'aggregate operation'.

WR10: The composite_signal_property_relationship shall reference exactly one characterized_object that is also an external_definition.

5.2.3.1.78 composite_unit_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A composite_unit_shape_aspect is a composite_shape_aspect that implements the ARM concept of Composite_unit_shape_element (see 4.2.167).

EXPRESS specification:

```

*)
ENTITY composite_unit_shape_aspect
  SUBTYPE OF (composite_shape_aspect);
END_ENTITY; -- composite_unit_shape_aspect
( *

```

5.2.3.1.79 concentricity_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A concentricity_tolerance is a geometric_tolerance_with_specified_datum_system that implements the ARM concept of Concentricity_and_coaxiality_tolerance (see 4.2.168).

EXPRESS specification:

```

*)
ENTITY concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'concentricity';
END_ENTITY; -- concentricity_tolerance
( *

```

Formal propositions:

WR1: The name for the concentricity_tolerance shall be 'concentricity'.

Informal propositions:

IP1: The tolerated_shape_aspect of the concentricity_tolerance shall either be a centre_of_symmetry with a description of either 'axis' or 'point'; or a composite_shape_aspect in which the 'composing' shape_aspects are all of type centre_of_symmetry and have the same description, either 'axis' or 'point'.

IP2: The concentricity_tolerance shall have exactly one physical_unit_datum in its referenced datum_system.

IP3: If the tolerated_shape_aspect of the concentricity_tolerance is either a centre_of_symmetry with a description of 'point', or a composite_shape_aspect in which the 'composing' shape_aspects are all of type centre_of_symmetry with a description of 'point'; the physical_unit_datum of the datum_system shall have a description of 'point'.

IP4: If the `toleranced_shape_aspect` of the `concentricity_tolerance` is either a `centre_of_symmetry` with a description of 'axis', or a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'axis'; the `physical_unit_datum` of the `datum_system` shall have a description of 'axis'.

IP5: The `tolerance_zone_boundary` of the `concentricity_tolerance` shall have a description of 'circular or cylindrical or spherical'.

5.2.3.1.80 `conductance_measure_with_unit`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `conductance_measure_with_unit` is a `measure_with_unit` that implements the ARM concept of `Conductance_measure_with_unit` (see ??).

EXPRESS specification:

```
*)
ENTITY conductance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: sizeof(typeof(self) - (typeof(self\measure_with_unit ||
    self\conductance_measure_with_unit))) = 0;

  WR2: self\measure_with_unit.unit_component\si_unit.name = siemens;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    typeof(self\measure_with_unit.unit_component);
END_ENTITY; -- conductance_measure_with_unit
(*
```

Formal propositions:

WR1: The `conductance_measure_with_unit` shall not be any other type of `measure_with_unit`.

WR2: The `conductance_measure_with_unit` shall reference an `si_unit` with the name of `siemens`.

WR3: The `unit_component` shall be an `si_unit`.

5.2.3.1.81 `conductive_interconnect_element_terminal_link`

A `conductive_interconnect_element_terminal_link` is a `shape_aspect` and a `shape_aspect_relationship` that implements the ARM concept of `Conductive_interconnect_element_terminal_link` (see 4.2.173).

EXPRESS specification:

```

*)
ENTITY conductive_interconnect_element_terminal_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relate_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL' IN
        TYPEOF (SELF\shape_aspect_relationship.relate_shape_aspect);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\conductive_interconnect_element_terminal_link)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';

  WR7: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF\shape_aspect.of_shape.definition);

  WR8: SELF\shape_aspect_relationship.related_shape_aspect.description = 'conductive interconnect_element_terminal';

  WR9: SELF\shape_aspect_relationship.relate_shape_aspect.description = 'conductive interconnect_element_terminal';

  WR10: SELF\shape_aspect.of_shape.definition\product_definition.description = 'laminate conductive interconnect_element_terminal';
END_ENTITY; -- conductive_interconnect_element_terminal_link
(

```

Formal propositions:

WR1: The related point shall not be the relating point.

WR2: The conductive_interconnect_element_terminal_link shall relate to only a conductive_interconnect_element_terminal in the related_shape_aspect.

WR3: The `conductive_interconnect_element_terminal_link` shall relate to only a `conductive_interconnect_element_terminal` in the `relating_shape_aspect`.

WR4: The types of `conductive_interconnect_element_terminal_link` shall be only `conductive_interconnect_element_terminal_link`, `shape_aspect`, and `shape_aspect_relationship`.

WR5: The name attribute of the `shape_aspect` supertype shall be an empty string.

WR6: The name attribute of the `shape_aspect_relationship` supertype shall be an empty string.

WR7: The `shape_aspect` supertype shall reference a `component_definition`

WR8: The related `shape_aspect` description shall be a 'conductive interconnect element terminal'.

WR9: The relating `shape_aspect` description shall be a 'conductive interconnect element terminal'.

WR10: The associated `product_definition` description shall be a 'laminated component'.

5.2.3.1.82 **connected_area_component**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `connected_area_component` is a `component_shape_aspect` that implements the ARM concept of `Connected_area_component` (see 4.2.183)

EXPRESS specification:

*)

```
ENTITY connected_area_component
  SUBTYPE OF (component_shape_aspect);
WHERE
```

```
WR1: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_DEFINITION' IN
      TYPEOF (SELF.of_shape.definition)) AND
      (SIZEOF (QUERY (prpc <*
        USEDIN (SELF.of_shape.definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
        prpc\product_category.name = 'template model')) >= 1);
```

```
WR2: SIZEOF (QUERY (sfi <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'stratum feature implementation') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM_FEATURE' IN
      TYPEOF (sfi.relying_shape_aspect))) = 1;
```

```

WR3: SIZEOF (QUERY (sfi <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated layer connection point') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYER_CONNECTION_POINT' IN
    TYPEOF (sfi.relateing_shape_aspect))) >= 1;

WR4: SELF\shape_aspect.description = 'stratum feature template component';

WR5: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\component_shape_aspect||
    SELF\connected_area_component)) = 0;
END_ENTITY; -- connected_area_component
(*

```

Formal propositions:

WR1: The `connected_area_component` shall be an aspect of the shape of a `component_definition` that is the definition of a product that is categorized as an 'template module'.

WR2: The `connected_area_component` shall be related to exactly one `stratum_feature` as a 'stratum feature implementation'.

WR3: The `connected_area_component` shall be related to at least one `layer_connection_point` as an 'associated layer connection point'.

WR4: The `connected_area_component` shall have a description of 'stratum feature template component'.

WR5: The `connected_area_component` shall not be a complex instance other than with its direct super-types.

5.2.3.1.83 connection_zone_based_assembly_joint

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `connection_zone_based_assembly_joint` is an `assembly_joint` that includes the identification of the connection zones that are to be mated in addition to the specific features that are to be mated. A `connection_zone_based_assembly_joint` implements the ARM concept of `Connection_zone_based_assembly_joint` (see 4.2.186).

EXPRESS specification:

```

*)
ENTITY connection_zone_based_assembly_joint
  SUBTYPE OF (assembly_joint);
WHERE

  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'zone 1')) = 1;

  WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'zone 2')) = 1;

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar.relating_shape_aspect\shape_aspect.description = 'connection zone')) = 2;

END_ENTITY; -- connection_zone_based_assembly_joint
(

```

Formal propositions:

WR1: The connection_zone_based_assembly_joint shall be related to exactly one shape_aspect as 'zone 1' by a shape_aspect_relationship.related_shape_aspect.

WR2: The connection_zone_based_assembly_joint shall be related to exactly one shape_aspect as 'zone 2' by a shape_aspect_relationship.relating_shape_aspect.

WR3: The connection_zone_based_assembly_joint shall be related to only shape_aspects with a description of 'connection_zone'.

5.2.3.1.84 connection_zone_based_fabrication_joint

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A connection_zone_based_fabrication_joint is an fabrication_joint that includes the identification of the connection zones that are to be mated in addition to the specific features that are to be mated. A connection_zone_based_fabrication_joint implements the ARM concept of Connection_zone_based_fabrication_joint (see 4.2.187).

EXPRESS specification:

```

*)
ENTITY connection_zone_based_fabrication_joint
  SUBTYPE OF (fabrication_joint);
WHERE

  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'zone 1')) = 1;

  WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'zone 2')) = 1;

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar.relating_shape_aspect\shape_aspect.description = 'connection zone')) = 2;

END_ENTITY; -- connection_zone_based_fabrication_joint
( *

```

Formal propositions:

WR1: The connection_zone_based_fabrication_joint shall be related to exactly one shape_aspect as 'zone 1' by a shape_aspect_relationship.related_shape_aspect.

WR2: The connection_zone_based_fabrication_joint shall be related to exactly one shape_aspect as 'zone 2' by a shape_aspect_relationship.relating_shape_aspect.

WR3: The connection_zone_based_fabrication_joint shall be related to only shape_aspects with a description of 'connection_zone'.

5.2.3.1.85 connection_zone_interface_plane_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A connection_zone_interface_plane_relationship is a shape_aspect and shape_aspect_relationship that implements the ARM concept of Connection_zone_interface_plane_relationship (see 4.2.188).

EXPRESS specification:

```

*)
ENTITY connection_zone_interface_plane_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
UNIQUE
  UR1: SELF\shape_aspect_relationship.name;
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect_relationship ||
    SELF\connection_zone_interface_plane_relationship ||
    SELF\shape_aspect)) = 0;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE' IN
    TYPEOF(SELF\shape_aspect_relationship.relate_shape_aspect);

  WR3: SELF\shape_aspect_relationship.relate_shape_aspect\shape_aspect.description =
    'plane';

  WR4: SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description =
    'connection zone';

  WR5: SELF\shape_aspect_relationship.description IN ['area', 'edge'];

  WR6: SELF\shape_aspect.of_shape = SELF\shape_aspect_relationship.related_shape_aspect.of_shape;

  WR7: SELF\shape_aspect.of_shape = SELF\shape_aspect_relationship.relate_shape_aspect.of_shape;

  WR8: SIZEOF (QUERY (dz <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATE_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'derived zone') |
    dz.related_shape_aspect\shape_aspect.description = 'connection zone')) = 1;
END_ENTITY; -- connection_zone_interface_plane_relationship
(

```

Formal propositions:

UR1: The name shall be unique.

WR1: A connection_zone_interface_plane_relationship shall not be any other type of shape_aspect_relationship or any other type of shape_aspect.

WR2: A connection_zone_interface_plane_relationship shall relate to a seating_plane.

WR3: A connection_zone_interface_plane_relationship shall relate to a seating_plane, with a name of 'plane'.

WR4: A connection_zone_interface_plane_relationship shall relate to a shape_aspect with a description of 'connection zone'.

WR5: A connection_zone_interface_plane_relationship shall have a description of 'area' or 'edge'.

WR6: A connection_zone_interface_plane_relationship shall have the same of_shape through the shape_aspect supertype and through the related_shape_aspect in the shape_aspect_relationship supertype.

WR7: A connection_zone_interface_plane_relationship shall have the same of_shape through the shape_aspect supertype and through the relating_shape_aspect in the shape_aspect_relationship supertype.

WR8: A connection_zone_interface_plane_relationship shall be related to one shape_aspect with a description of 'connection zone' as the 'derived zone'.

5.2.3.1.86 connection_zone_map_identification

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A connection_zone_map_identification is a shape_aspect and a representation_relationship that implements the ARM concept of Connection_zone_map_identification (see 4.2.189)

EXPRESS specification:

*)

```
ENTITY connection_zone_map_identification
  SUBTYPE OF (shape_aspect, representation_relationship);
WHERE
```

```
  WR1: SELF\representation_relationship.rep_2 :<>:
        SELF\representation_relationship.rep_1;
```

```
  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'USAGE_VIEW_CONNECTION_ZONE_TERMINAL_SHAPE_RELATIONSHIP' IN
  TYPEOF(SELF\representation_relationship.rep_1);
```

```
  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'USAGE_VIEW_CONNECTION_ZONE_TERMINAL_SHAPE_RELATIONSHIP' IN
  TYPEOF(SELF\representation_relationship.rep_2);
```

```
  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\representation_relationship ||
        SELF\connection_zone_map_identification)) = 0;
```

```
  WR5: SELF\shape_aspect.name = '';
```

```

WR6: SELF\representation_relationship.name = '';

WR7: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FOOTPRINT_DEFINITION' IN TYPEOF(SELF.of_shape.definition);
END_ENTITY; -- connection_zone_map_identification
(*)

```

Formal propositions:

- WR1:** The related zone shall not be the relating zone.
- WR2:** The connection_zone_map_identification shall relate to only a usage_view_connection_zone-terminal_shape_relationship in the rep_1.
- WR3:** The connection_zone_map_identification shall relate to only a usage_view_connection_zone-terminal_shape_relationship in the rep_2.
- WR4:** The types of connection_zone_map_identification shall be only connection_zone_map_identification, shape_aspect, and representation_relationship.
- WR5:** The name attribute of the shape_aspect supertype shall be an empty string.
- WR6:** The name attribute of the representation_relationship supertype shall be an empty string.
- WR7:** Each connection_zone_map_identification shall be a shape_aspect for a footprint_definition.

5.2.3.1.87 coordinated_representation_item

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A coordinated_representation_item is a representation and a representation_item that implements the ARM concepts of:

- Plus_minus_tolerance_characteristic (see 4.2.691);
- Statistical_tolerance_characteristic (see 4.2.842);
- Symmetrical_tolerance_characteristic (see 4.2.882);
- Tolerance_characteristic (see 4.2.911).

EXPRESS specification:

```

*)
ENTITY coordinated_representation_item
  SUBTYPE OF (representation, representation_item);
WHERE

```

```

WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    SIZEOF (USEDIN (pdr,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATA_ENVIRONMENT.ELEMENTS')) <= 1)) <= 1;

WR2: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;

WR3: (NOT (SELF\representation_item.name = 'tolerance')) OR
    (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM' IN
    TYPEOF (it))) >= 1);

WR4: (NOT (SELF\representation_item.name = 'plus minus tolerance')) OR
    (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM' IN
    TYPEOF (it))) = 3);

WR5: (NOT (SELF\representation_item.name = 'symmetrical tolerance')) OR
    (SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM' IN
    TYPEOF (it))) = 2);

```

```

WR6: (NOT (SELF\representation_item.name = 'plus minus tolerance'))
OR
(
  (SIZEOF (QUERY (it <* SELF.items
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it))
    AND
    (it\representation_item.name = 'basic value'))
  ) = 1)
AND
  (SIZEOF (QUERY (it <* SELF.items
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it))
    AND
    (it\representation_item.name = 'plus value'))
  ) = 1)
AND
  (SIZEOF (QUERY (it <* SELF.items
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it))
    AND
    (it\representation_item.name = 'minus value'))
  ) = 1)
);

WR7: (NOT (SELF\representation_item.name = 'symmetrical tolerance')) OR
((SIZEOF (QUERY (it <* SELF.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND (it\representation_item.name = 'basic value'))
) = 1)
AND
(SIZEOF (QUERY (it <* SELF.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND ( it\representation_item.name = 'deviation value'))
) = 1)
);
END_ENTITY; -- coordinated_representation_item
( *

```

Formal propositions:

WR1: The coordinated_representation_item shall represent at most one property that is an element of at most one data_environment.

WR2: The coordinated_representation_item shall be referenced by exactly one document.

WR3: If the `coordinated_representation_item` has a `representation_item` name of 'tolerance', it shall contain at least one `measure_representation_item`.

WR4: If the `coordinated_representation_item` has a `representation_item` name of 'plus minus tolerance', it shall contain exactly three `measure_representation_items`.

WR5: If the `coordinated_representation_item` has a `representation_item` name of 'symmetrical tolerance', it shall contain exactly two `measure_representation_items`.

WR6: If the `coordinated_representation_item` has a `representation_item` name of 'plus minus tolerance', the three contained `measure_representation_items` shall have their `name` attribute satisfied by 'basic value', 'plus value', and 'minus value'.

WR7: If the `coordinated_representation_item` has a `representation_item` name of 'symmetrical tolerance', the two contained `measure_representation_items` shall have their `name` attribute satisfied by 'basic value' and 'deviation value'.

Informal propositions:

IP1: If the `coordinated_representation_item` has a `representation_item` name of 'plus minus tolerance', each contained `measure_representation_items` shall have the same `unit_components` for the `dimensions` attribute.

IP2: If the `coordinated_representation_item` has a `representation_item` name of 'plus minus tolerance', the `measure_representation_item` with a `name` attribute of 'minus value' shall have a `value_component` less than that of the `measure_representation_item` with a `name` attribute of 'basic value' which shall have a `value_component` less than that of the `measure_representation_item` with a `name` attribute of 'plus_value'.

5.2.3.1.88 `csg_2d_shape_representation`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `csg_2d_shape_representation` is a `shape_representation` that implements the ARM concept of `Area_2d_shape_definition` (see ??). A `csg_2d_shape_representation` is a two-dimensional `shape_representation` that represents the shape of a product using constructive solid geometry operations and techniques. The solid objects may be either 2D primitives, or half-space 2D solids.

EXPRESS specification:

```
* )
ENTITY csg_2d_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
```

```

WR1: SELF.context_of_items\
    geometric_representation_context.coordinate_space_dimension = 2;

WR2 : SIZEOF (QUERY (it <* SELF.items| NOT
    (SIZEOF(['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.CSG_SOLID',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.MAPPED_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.AXIS2_PLACEMENT_2D']
    * TYPEOF(it)) = 1 ))) = 0;

WR3: SIZEOF(QUERY ( it <* SELF.items |
    (SIZEOF(['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.CSG_SOLID']
    * TYPEOF(it)) = 1)
OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.MAPPED_ITEM'
    IN TYPEOF(it)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CSG_2D_SHAPE_REPRESENTATION'
    IN TYPEOF(it\mapped_item.mapping_source.mapped_representation)))
    )) >= 1;

WR4: SIZEOF(QUERY ( it <* SELF.items |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.MAPPED_ITEM'
    IN TYPEOF(it)) AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CSG_2D_SHAPE_REPRESENTATION')
    IN TYPEOF(it\mapped_item.mapping_source.mapped_representation)))))) = 0;

WR5: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_representation ||
    SELF\csg_2d_shape_representation || SELF\representation)) = 0;
END_ENTITY; -- csg_2d_shape_representation
(*

```

Formal propositions:

WR1: The `geometric_representation_context` of the `csg_2d_shape_representation` shall have a `coordinate_space_dimension` equal to two.

WR2: The items in a `csg_2d_shape_representation` shall be a `csg_solid`, `mapped_item`, or an `axis2_placement_2d`.

WR3: At least one of the items in a `csg_2d_shape_representation` shall be a `csg_solid`, or a `mapped_item` with a `mapped_representation` of the `mapping_source` attribute of the type `csg_2d_shape_representation`.

WR4: If a `mapped_item` is in the items set then the `mapped_representation` of the `mapping_source` attribute shall be a `csg_2d_shape_representation`.

WR5: The `csg_2d_shape_representation` shall not be another type of `shape_representation`.

NOTE Since WR1 requires a coordinate_space_dimension equal to two, the leaves of the tree_root_expression in csg_solid can be built by only half_space_2d or primitive_2d elements.

5.2.3.1.89 curve_dimension

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A curve_dimension is a dimensional_size that implements the ARM concept of Curve_dimension (see 4.2.204).

EXPRESS specification:

```
*)
ENTITY curve_dimension
  SUBTYPE OF (dimensional_size);
WHERE

  WR1: SELF\dimensional_size.name <> 'angular' ;
END_ENTITY; -- curve_dimension
(*
```

Formal propositions:

WR1: The name of the curve_dimension shall not be 'angular'.

5.2.3.1.90 cutout_edge_segment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A cutout_edge_segment is an inter_stratum_feature and a shape_aspect_relationship that implements the ARM concept of Cutout_edge_segment (see 4.2.209).

EXPRESS specification:

```
*)
ENTITY cutout_edge_segment
  SUPERTYPE OF (plated_cutout_edge_segment)
  SUBTYPE OF (inter_stratum_feature, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect.description IN [
    'plated cutout edge segment',
    'cutout edge segment'];
```

```

WR2: sizeof (QUERY (cc <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'composed cutout') |(
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTER_STRATUM_FEATURE' IN
    TYPEOF (cc.relatng_shape_aspect)) AND
(cc.relatng_shape_aspect\shape_aspect.description IN ['cutout',
    'physical connectivity interrupting cutout', 'plated cutout']))
    )) = 1;

WR3: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.relatng_shape_aspect));

WR4: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect));

WR5: SELF\shape_aspect_relationship.related_shape_aspect :<:
    SELF\shape_aspect_relationship.relatng_shape_aspect;
END_ENTITY; -- cutout_edge_segment
(*

```

Formal propositions:

WR1: The cutout_edge_segment shall have a name of 'cutout edge segment' and a relating_shape_aspect_description of 'cutout', 'physical connectivity interrupting cutout', or 'plated cutout'. 'plated cutout edge segment'.

WR2: The cutout_edge_segment shall relate to an edge_segment_vertex in the relating role.

WR3: The cutout_edge_segment shall relate to an edge_segment_vertex in the related role.

WR4: The relating_shape_aspect and the related_shape_aspect shall not be the same instance of shape_aspect.

Informal propositions:

IP1: The associated passage_technology shall have descriptive_representation_item with a description of 'true'.

5.2.3.1.91 data_set_representation_item

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A data_set_representation_item is a compound_representation_item that implements the ARM concept of Characteristic_data_set (see 4.2.99).

EXPRESS specification:

```

*)
ENTITY data_set_representation_item
  SUBTYPE OF (compound_representation_item);
WHERE

  WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS')) > 0;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SET_REPRESENTATION_ITEM'
    IN TYPEOF(SELF\compound_representation_item.item_element);
END_ENTITY; -- data_set_representation_item
( *

```

Formal propositions:

WR1: The data_set_representation_item shall be referenced by exactly one applied_document_reference.

WR2: The item_element attribute shall be a list_representation_item.

5.2.3.1.92 datum_difference

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_difference is a subtype of shape_aspect and shape_aspect_relationship that implements the ARM concept of Datum_difference (see 4.2.221)

EXPRESS specification:

```

*)
ENTITY datum_difference
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM' IN TYPEOF
    (SELF\shape_aspect_relationship.relate_shape_aspect)) AND
    ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM' IN TYPEOF
    (SELF\shape_aspect_relationship.related_shape_aspect));

```

```

WR2: SELF\shape_aspect.name = SELF\shape_aspect_relationship.name;

WR3: SELF\shape_aspect.description = SELF\shape_aspect_relationship.
      description;

WR4: SELF\shape_aspect_relationship.relate_shape_aspect :<>:
      SELF\shape_aspect_relationship.related_shape_aspect;

WR5: SIZEOF (TYPEOF(SELF) - (TYPEOF(SELF\shape_aspect ||
      SELF\shape_aspect_relationship || SELF\datum_difference))) = 0;
END_ENTITY; -- datum_difference
(*)

```

Formal propositions:

- WR1:** The datum_difference shall be applied to only physical_unit_datum.
- WR2:** The datum_difference name attribute inherited shall be identical in each supertypes.
- WR3:** The datum_difference description attribute inherited shall be identical in each supertypes.
- WR4:** The datum_difference shall be applied to only different physical_unit_datum.
- WR5:** The datum_difference shall not be in a complex instance of other types of shape_aspect or shape_aspect_relationship.

5.2.3.1.93 datum_difference_analytical_model_port_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_difference_analytical_model_port_assignment is a subtype of property_definition_representation that implements the ARM concept of Datum_difference_analytical_model_port_assignment (see 4.2.222)

EXPRESS specification:

```

*)
ENTITY datum_difference_analytical_model_port_assignment
  SUBTYPE OF (property_definition_representation);
WHERE

```

```

WR1: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANALYTICAL_MODEL_PORT' IN TYPEOF
      (SELF\property_definition_representation.used_representation)) AND
      ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE' IN TYPEOF
      (SELF\property_definition_representation.definition));

WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\property_definition_representation ||
      SELF\datum_difference_analytical_model_port_assignment)) = 0;
END_ENTITY; -- datum_difference_analytical_model_port_assignment
(*)

```

Formal propositions:

WR1: The datum_difference_analytical_model_port_assignment shall be related to only datum_difference and analytical_model_port.

WR2: The datum_difference_analytical_model_port_assignment shall not be in a complex instance of other types of property_definition_representation.

5.2.3.1.94 datum_difference_based_characteristic

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_difference_based_characteristic is a representation_item that implements the ARM concept of Datum_difference_based_characteristic (see 4.2.223)

EXPRESS specification:

```

*)
ENTITY datum_difference_based_characteristic
  SUBTYPE OF (representation_item);
WHERE

```

```

WR1: sizeof(QUERY ( r <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REPRESENTATION.ITEMS') | (sizeof(QUERY (prd2 <*
    QUERY ( prd <* USEDIN(r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
      ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
      + 'PROPERTY_DEFINITION_REPRESENTATION') IN TYPEOF(prd)) AND
      (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION') IN TYPEOF(prd\
      property_definition_representation.definition))) ) |
      ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE' IN
      TYPEOF(prd2\property_definition_representation.definition\
      property_definition.definition))))
      = 1) ))
      = 1;
END_ENTITY; -- datum_difference_based_characteristic
( *

```

Formal propositions:

WR1: The datum_difference_based_characteristic shall be related to exactly one datum_difference.

5.2.3.1.95 datum_difference_based_model_parameter

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_difference_based_model_parameter is a model_parameter that implements the ARM concept of Datum_difference_based_model_parameter (see 4.2.224).

EXPRESS specification:

```

*)
ENTITY datum_difference_based_model_parameter
  SUBTYPE OF (model_parameter);
WHERE

```

```

WR1: SIZEOF(QUERY ( r <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | (SIZEOF(QUERY (prd2 <*
    QUERY ( prd <* USEDIN(r,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'PROPERTY_DEFINITION_REPRESENTATION') IN TYPEOF(prd)) AND
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION') IN TYPEOF(prd\
    property_definition_representation.definition))) ) |
    ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_DIFFERENCE' IN
    TYPEOF(prd2\property_definition_representation.definition\
    property_definition.definition))))
    = 1) ))
    = 1;
END_ENTITY; -- datum_difference_based_model_parameter
(*)

```

Formal propositions:

WR1: The datum_difference_based_model_parameter shall be related to exactly one datum_difference.

5.2.3.1.96 datum_difference_functional_unit_usage_view_terminal_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_difference_functional_unit_usage_view_terminal_assignment is a shape_aspect_relationship that implements the ARM concept of Datum_difference_functional_unit_usage_view_terminal_assignment (see 4.2.225).

EXPRESS specification:

```

*)
ENTITY datum_difference_functional_unit_usage_view_terminal_assignment
    SUBTYPE OF (shape_aspect_relationship);
WHERE

```

```

    WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect_relationship ||
    SELF\datum_difference_functional_unit_usage_view_terminal_assignment))
        = 0;

```

```

    WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' IN TYPEOF
    (SELF\shape_aspect_relationship.related_shape_aspect);

```

```

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE' IN TYPEOF
      (SELF\shape_aspect_relationship.relate_shape_aspect);
END_ENTITY; -- datum_difference_functional_unit_
--          usage_view_terminal_assignment
(*)

```

Formal propositions:

WR1: The datum_difference_functional_unit_usage_view_terminal_assignment shall not participate in a complex instance with other than the shape_aspect_relationship supertype.

WR2: The datum_difference_functional_unit_usage_view_terminal_assignment shall be related to exactly one functional_unit_terminal_definition.

WR3: The datum_difference_functional_unit_usage_view_terminal_assignment shall be related to exactly one datum_difference.

5.2.3.1.97 datum_reference_frame

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_reference_frame is a shape_aspect that implements the ARM concept of Datum_reference_frame (see 4.2.235).

EXPRESS specification:

```

*)
ENTITY datum_reference_frame
  SUBTYPE OF (shape_aspect);
WHERE

```

```

WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
  SELF\datum_reference_frame)) = 0;

```

```

WR2: SELF\shape_aspect.product_definitional = False;

```

```

WR3: (SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'reference axis') |
  (pud.related_shape_aspect\shape_aspect.description = 'axis') AND
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect)))) <= 3);

```



```

WR4: (SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference plane') |
(pud.related_shape_aspect\shape_aspect.description = 'plane') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect)))) <= 3);

WR5: (SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference origin') |
(pud.related_shape_aspect\shape_aspect.description = 'point') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect)))) <= 1);

WR6: (SIZEOF (QUERY (ds <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'established datum reference frame') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_SYSTEM' IN TYPEOF (ds.related_shape_aspect))) >= 1);

WR7: (SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference axis') |
(pud.related_shape_aspect\shape_aspect.description = 'axis') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect))))
+ SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference plane') |
(pud.related_shape_aspect\shape_aspect.description = 'plane') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect))))
+ SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference origin') |
(pud.related_shape_aspect\shape_aspect.description = 'point') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN TYPEOF (pud.related_shape_aspect)))) >= 1);

```

```

WR8: (SIZEOF (QUERY (pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'reference origin') |
      (pud.related_shape_aspect\shape_aspect.description <> 'point') OR
      (NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DERIVED_SHAPE_ASPECT' IN TYPEOF (pud.related_shape_aspect)))
      )) = 0);
END_ENTITY; -- datum_reference_frame
(*

```

Formal propositions:

WR1: The datum_reference_frame shall not be a complex instance, except with the supertype shape_aspect.

WR2: The datum_reference_frame shall not lie on the boundary of the shape of the product.

WR3: The datum_reference_frame shall relate to at most three physical_unit_datum that have a description of 'axis' to specify the 'reference axis'.

WR4: The datum_reference_frame shall relate to at most three physical_unit_datum that have a description of 'plane' to specify the 'reference plane'.

WR5: The datum_reference_frame shall relate to at most one physical_unit_datum that has a description of 'point' to specify the 'reference origin'.

WR6: The datum_reference_frame shall be related to at least one datum_system as the 'established datum reference frame'.

WR7: The datum_reference_frame shall relate to at least one physical_unit_datum that has a description of 'axis' or 'plane' or 'point' to specify the 'reference axis' or 'reference plane' or 'reference origin', respectively.

WR8: Each 'reference origin' related to the datum_reference_frame shall also be a derived_shape_aspect that has a description of 'point'.

Informal propositions:

IP1: Each reference_axis shall be specified to be either a constituent of, or normal to each of the reference_planes and this specification shall be consistent with the mutually perpendicular reference_planes.

5.2.3.1.98 datum_system

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_system is a shape_aspect that implements the ARM concept of Datum_system (see 4.2.236).

EXPRESS specification:

```

*)
ENTITY datum_system
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\shape_aspect.product_definitional = False;

  WR2: (SIZEOF (QUERY (sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name =
      'established datum reference frame')) = 1);

  WR3: (SIZEOF (QUERY (sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name =
      'datum usage in datum system')) >= 1);

  WR4: (SIZEOF (QUERY (pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (pd\property_definition.description =
      'datum system property with material conditions')
    OR (pd\property_definition.description =
      'datum system property without material conditions')))) = 1);
END_ENTITY; -- datum_system
(

```

Formal propositions:

WR1: The datum_system shall not lie on the boundary of the shape of the product.

WR2: The datum_system shall relate to at exactly one shape_aspect to specify the 'established datum reference frame'.

WR3: The datum_system shall relate to at least one shape_aspect to specify the 'datum usage in datum system'.

WR4: The datum_system shall have exactly one property_definition that has the description of either 'datum system property with material conditions' or 'datum system property without material conditions'.

Informal propositions:

IP1: Any particular physical_unit_datum_feature shall be used no more than one time to establish the datum_system.

IP2: Any particular physical_unit_datum_target shall be used no more than one time to establish the datum_system.

5.2.3.1.99 datum_system_based_dimensional_location

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A datum_system_based_dimensional_location is a dimensional_location that implements the ARM concept of Dimension_location_with_specified_datum_system (see 4.2.293).

EXPRESS specification:

```
*)
ENTITY datum_system_based_dimensional_location
    SUBTYPE OF (dimensional_location);
END_ENTITY; -- datum_system_based_dimensional_location
(*
```

Informal propositions:

IP1: The datum_system_based_dimensional_location shall have exactly one property_definition with a description of 'dimensional location property', and that property_definition shall relate to a property_definition with a description of 'datum system property without material conditions' to specify the 'referenced datum system property'.

5.2.3.1.100 delete_design_object_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A delete_design_object_assignment is an action_assignment that implements the ARM concept of Delete_design_object_management_relationship (see 4.2.257).

EXPRESS specification:

```

*)
ENTITY delete_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

  WR1: SIZEOF(QUERY(it <* SELF.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(it))
    AND (it\product_definition_relationship.name = 'design object deletion')) = 1;
END_ENTITY; -- delete_design_object_assignment
(*

```

Formal propositions:

WR1: The delete_design_object_assignment shall have exactly one item that is a product_definition_relationship with a name of 'design object deletion'.

Informal propositions:

IP1: The delete_design_object_assignment shall have exactly one item that is a managed_design_object that is not a product_definition_relationship with a name of 'design object deletion' and that is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design.

5.2.3.1.101 delete_design_object_request_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A delete_design_object_request_assignment is an action_request_assignment that implements the ARM concepts of Change_request (see 4.2.97) and Delete_design_object_management_relationship (see 4.2.257).

EXPRESS specification:

```

*)
ENTITY delete_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE

```

```

WR1: sizeof(QUERY(it <* SELF.items |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(it))
  AND (it\product_definition_relationship.name = 'design object deletion')) = 1;
END_ENTITY; -- delete_design_object_request_assignment
(*

```

Formal propositions:

WR1: The delete_design_object_request_assignment shall have exactly one item that is a product_definition_relationship with a name of 'design object deletion'.

Informal propositions:

IP1: The delete_design_object_request_assignment shall have exactly one item that is a managed_design_object that is not a product_definition_relationship with a name of 'design object deletion' and that is not referenced by another product_definition_relationship with a name of 'design object addition', 'design object change', or 'design object deletion' in the same design.

5.2.3.1.102 dependent_material_removal_feature_template

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A dependent_material_removal_feature_template is a part_template_definition that implements the ARM concept of Dependent_material_removal_feature_template (see 4.2.258)

EXPRESS specification:

```

*)
ENTITY dependent_material_removal_feature_template
  SUBTYPE OF (part_template_definition);
WHERE

```

```

WR1: SELF\shape_aspect.description = 'material removal feature template';

```

```

WR2: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated template') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND_PHYSICAL_TEMPLATE' IN
    TYPEOF (am.relate_shape_aspect)) AND
    (am.relate_shape_aspect.description IN
    ['default attachment size and component termination passage based',
    'default attachment size based',
    'default attachment size and via based',
    'default component termination passage based',
    'default via based',
    'default unsupported passage based'])))
    )) = 1;
END_ENTITY; -- dependent_material_removal_feature_template
(*

```

Formal propositions:

WR1: The `dependent_material_removal_feature_template` shall have a description that is 'material removal feature template'.

WR2: The `dependent_material_removal_feature_template` shall be related to exactly one `land_physical_template` with a description of 'default attachment size and component termination passage based', or 'default attachment size based', or 'default attachment size and via based', or 'default component termination passage based', or 'default via based', or 'default unsupported passage based'.

5.2.3.1.103 design_layer_type_specific_padstack_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `design_layer_type_specific_padstack_definition` is a `product_definition` that implements the ARM concepts of: `Design_layer_type_specific_padstack_definition` (see ??)

EXPRESS specification:

```

*)
ENTITY design_layer_type_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- design_layer_type_specific_padstack_definition
(*

```

5.2.3.1.104 design_make_from_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `design_make_from_relationship` is a `product_definition_relationship` that implements the ARM concept of `Engineering_make_from` (see 4.2.353).

EXPRESS specification:

```

*)
ENTITY design_make_from_relationship
  SUBTYPE OF (product_definition_relationship);
WHERE

  WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' ] *
    TYPEOF (SELF.relying_product_definition)) = 1;

  WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' ] *
    TYPEOF (SELF.related_product_definition)) = 1;

  WR3: SIZEOF( QUERY(pds <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)))) = 0;
END_ENTITY; -- design_make_from_relationship
( *
```

Formal propositions:

WR1: The `design_make_from_relationship` shall reference a `physical_unit` or a `functional_unit` as its `relying_product_definition`.

WR2: The `design_make_from_relationship` shall reference a `physical_unit` or a `functional_unit` as its `related_product_definition`.

WR3: The `design_make_from_relationship` shall not be the definition of a `product_definition_shape`.

5.2.3.1.105 **design_object**

A `design_object` is a `characterized_object` that implements the ARM concept of `Layout_spacing_requirement_occurrence` (see 4.2.509).

EXPRESS specification:

```

*)
ENTITY design_object
  SUBTYPE OF (characterized_object);
END_ENTITY; -- design_object
( *

```

5.2.3.1.106 device_terminal_map

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A device_terminal_map is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Device_terminal_map (see 4.2.285).

EXPRESS specification:

```

*)
ENTITY device_terminal_map
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
UNIQUE
  UR1: SELF\shape_aspect_relationship.related_shape_aspect,
        SELF\shape_aspect_relationship.relying_shape_aspect;
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART_TERMINAL'
        IN TYPEOF (SELF.relying_shape_aspect);
END_ENTITY; -- device_terminal_map
( *

```

Formal propositions:

UR1: The combination of related_shape_aspect and relating_shape_aspect of shape_aspect_relationship shall be unique within a population of device_terminal_map.

WR1: The relating_shape_aspect shall be a packaged_part_terminal.

5.2.3.1.107 dimensional_location_with_direction

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A dimensional_location_with_direction is a dimensional_location that implements the ARM concept of Dimension_location_with_direction_vector (see 4.2.292).

EXPRESS specification:

```

*)
ENTITY dimensional_location_with_direction
  SUBTYPE OF (dimensional_location);
WHERE

  WR1: SELF\shape_aspect_relationship.description = 'linear';
END_ENTITY; -- dimensional_location_with_direction
( *

```

Formal propositions:

WR1: The description of the `dimensional_location_with_direction` shall be 'linear'.

Informal propositions:

IP1: The `dimensional_location_with_direction` shall have exactly one `property_definition` with a description of 'dimensional location property', and that `property_definition` either shall be represented by a representation that contains a placement with a name of 'orientation' or shall relate to a `property_definition` with a description of 'datum based vector orientation' to specify the 'measurement orientation'.

IP2: If the `dimensional_location_with_direction` has a `property_definition` with a description of 'dimensional location property' and that `property_definition` relates to a `property_definition` with a description of 'datum based vector orientation' to specify the 'measurement orientation', the combination of the `shape_aspect` specified as the `related_shape_aspect`, the `shape_aspect` specified as the `relating_shape_aspect`, and the `property_definition` with a description of 'datum based vector orientation' shall be unique.

IP3: If the `dimensional_location_with_direction` has a `property_definition` with a description of 'dimensional location property' and that `property_definition` is represented by a representation that contains a placement with a name of 'orientation', the combination of the `shape_aspect` specified as the `related_shape_aspect`, the `shape_aspect` specified as the `relating_shape_aspect`, and the placement with a name of 'orientation' shall be unique.

5.2.3.1.108 dimensional_size_property

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `dimensional_size_property` is a `dimensional_size` and a `property_definition`.

EXPRESS specification:

```

*)
ENTITY dimensional_size_property
  SUBTYPE OF (dimensional_size, property_definition);
END_ENTITY; -- dimensional_size_property
( *

```

5.2.3.1.109 directed_dimensional_location

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A directed_dimensional_location is a dimensional_location that implements the ARM concept of Dimensional_location (see 4.2.300) where the ARM attribute "directed" of that concept has a value of TRUE.

EXPRESS specification:

```

*)
ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY; -- directed_dimensional_location
( *

```

5.2.3.1.110 discrete_shield

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A discrete_shield is a component_definition that implements the ARM concept of Shield (see 4.2.821) when the shield is comprised of a component or components that are manufactured separately from the interconnect component.

EXPRESS specification:

```

*)
ENTITY discrete_shield
  SUBTYPE OF (component_definition);
WHERE

  WR1: SELF.frame_of_reference.name = 'physical occurrence';

```

```

WR2: SIZEOF (QUERY (si <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'shielded item') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (si.relatng_product_definition))) >= 1;
END_ENTITY; -- discrete_shield
(*

```

Formal propositions:

WR1: The discrete_shield shall have a frame_of_reference name of 'physical occurrence'.

WR2: The discrete_shield shall be related to at least one component_definition or at least one component_shape_aspect as the 'shielded item'.

5.2.3.1.111 document_identifier

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A document_identifier is a group that specifies an identifier qualified by a name for a document.

EXPRESS specification:

```

*)
ENTITY document_identifier
  SUBTYPE OF (group);
UNIQUE
  UR1: SELF\group.name, SELF\group.description;
WHERE

  WR1: SIZEOF (USEDIN(SELF, '')) = 1;
END_ENTITY; -- document_identifier
(*

```

Formal propositions:

UR1: The combination of name and description inherited from group shall be unique.

WR1: A document_identifier shall be referenced exactly once.

5.2.3.1.112 document_identifier_assignment

A document_identifier_assignment is a group_assignment that specifies one or more of document_identifier_assigned_item. If there are more than one, then the items are all revisions of the same drawing.

EXAMPLE 1 A schematic diagram may have a base identifier and each revision identification of that diagram differs from other revisions identification only in the document.description.

NOTE Due to a constraint on document_identifier, no member of document_identifier may be referenced more than once, guaranteeing uniqueness of assigned_group.

EXPRESS specification:

```
*)
ENTITY document_identifier_assignment
  SUBTYPE OF (group_assignment);
  SELF\group_assignment.assigned_group : document_identifier;
  items : SET [1:?] OF document_identifier_assigned_item;
END_ENTITY; -- document_identifier_assignment
(*
```

5.2.3.1.113 dose_equivalent_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A dose_equivalent_measure_with_unit is a measure_with_unit that implements the ARM concept of Dose_equivalent_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY dose_equivalent_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\dose_equivalent_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = sievert;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- dose_equivalent_measure_with_unit
(*
```

Formal propositions:

WR1: The dose_equivalent_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The dose_equivalent_measure_with_unit shall reference an si_unit with the name of sievert.

WR3: The unit_component shall be an si_unit.

5.2.3.1.114 edge_based_2d_wireframe_shape_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An edge_based_2d_wireframe_shape_representation is a shape_representation that implements the ARM concepts of Edge_based_2d_wireframe (see 4.2.309) and Edge_based_wireframe (see 4.2.310).

EXPRESS specification:

```

*)
ENTITY edge_based_2d_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
WHERE

  WR1: SIZEOF (QUERY (it <* SELF\representation.items |
    NOT (SIZEOF (
      ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_2D'] *
      TYPEOF (it)) = 1))) = 0;

  WR2: SIZEOF (QUERY (it <* SELF\representation.items |
    SIZEOF(['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM'] *
      TYPEOF (it)) = 1)) >= 1;

```

```

WR3: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_BASED_WIREFRAME_MODEL' IN
TYPEOF (it)) |
NOT (SIZEOF (QUERY (eb <*
ebwm\edge_based_wireframe_model.ebwm_boundary |
NOT (SIZEOF (QUERY (edges <* eb.ces_edges |
NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_CURVE'
IN TYPEOF (edges)))) = 0))) = 0))) = 0;

WR4: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_BASED_WIREFRAME_MODEL' IN
TYPEOF (it)) |
NOT (SIZEOF (QUERY (eb <*
ebwm\edge_based_wireframe_model.ebwm_boundary |
NOT (SIZEOF (QUERY (pline_edges <* QUERY (edges <* eb.ces_edges |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE' IN
TYPEOF (edges\edge_curve.edge_geometry)) |
NOT (SIZEOF (pline_edges\edge_curve.edge_geometry\polyline.points)
> 2))) = 0))) = 0))) = 0;

WR5: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_BASED_WIREFRAME_MODEL' IN
TYPEOF (it)) |
NOT (SIZEOF (QUERY (eb <*
ebwm\edge_based_wireframe_model.ebwm_boundary |
NOT (SIZEOF (QUERY (edges <* eb.ces_edges |
NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
TYPEOF (edges.edge_start)) AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
TYPEOF (edges.edge_end)))))) = 0))) = 0))) = 0;

WR6: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_BASED_WIREFRAME_MODEL' IN
TYPEOF (it)) |
NOT (SIZEOF (QUERY (eb <*
ebwm\edge_based_wireframe_model.ebwm_boundary |
NOT (SIZEOF (QUERY (edges <* eb.ces_edges |
NOT (valid_2d_wireframe_edge_curve
(edges\edge_curve.edge_geometry,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN'))))
= 0))) = 0))) = 0;

```

```

WR7: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'EDGE_BASED_WIREFRAME_MODEL' IN
  TYPEOF (it)) |
  NOT (SIZEOF (QUERY (eb <*
    ebwm\edge_based_wireframe_model.ebwm_boundary |
    NOT (SIZEOF (QUERY (edges <* eb.ces_edges |
      NOT ((valid_wireframe_vertex_point
        (edges.edge_start\vertex_point.vertex_geometry)) AND
        (valid_wireframe_vertex_point
          (edges.edge_end\vertex_point.vertex_geometry)))))) = 0))) = 0))) = 0;

WR8: SIZEOF (QUERY (ebwm <* QUERY (it <* SELF\representation.items |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'EDGE_BASED_WIREFRAME_MODEL' IN
  TYPEOF (it)) |
  NOT (SIZEOF (QUERY (eb <*
    ebwm\edge_based_wireframe_model.ebwm_boundary |
    NOT (SIZEOF (QUERY (con_edges <* QUERY (edges <* eb.ces_edges |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONIC'
      IN TYPEOF (edges\edge_curve.edge_geometry))) |
      NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D' IN
        TYPEOF (con_edges\edge_curve.edge_geometry\conic.position))))
      = 0))) = 0))) = 0;

WR9: SIZEOF (QUERY (mi <* QUERY (it <* SELF\representation.items |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MAPPED_ITEM' IN
  TYPEOF (it)) |
  NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION' IN
    TYPEOF (mi\mapped_item.mapping_source.mapped_representation)))) = 0;

WR10: SELF\representation.
  context_of_items\geometric_representation_context.
  coordinate_space_dimension = 2;
END_ENTITY; -- edge_based_2d_wireframe_shape_representation
(*

```

Formal propositions:

WR1: The items in an edge_based_2d_wireframe_shape_representation shall be an edge_based_wireframe_model, a mapped_item, or an axis2_placement_2d.

WR2: At least one of the items in an edge_based_2d_wireframe_shape_representation shall be either an edge_based_wireframe_model or a mapped_item.

WR3: Every edge in an `edge_based_wireframe_model` shall be an `edge_curve`.

WR4: Every polyline that underlies an edge of an `edge_based_wireframe_model` shall be defined by more than two distinct points.

WR5: Every vertex defined for an `edge_based_wireframe_model` shall be a `vertex_point`.

WR6: The `edge_geometry` that underlies an edge for an `edge_based_wireframe_model` shall be a line, circle, ellipse, parabola, hyperbola, `b_spline_curve`, `offset_curve_2d`, `polyline`, or a `curve_replica`. For each curve that has a basis defined by another curve, the basis curve shall be one of these types also.

WR7: The `vertex_geometry` that underlies the edges for an `edge_based_wireframe_model` shall be a `cartesian_point` or `point_replica` and the `point_replica` shall replicate either another `point_replica` or a `cartesian_point`.

WR8: All conics shall be positioned by an `axis2_placement_2d`.

WR9: If there is a `mapped_item` in an `edge_based_2d_wireframe_shape_representation`, the source of the `mapped_item` shall be an `edge_based_2d_wireframe_shape_representation`.

WR10: The `edge_based_2d_wireframe_shape_representation` shall have a `coordinate_space_dimension` equal to two.

5.2.3.1.115 `edge_segment_cross_section`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `edge_segment_cross_section` is a `volume_shape_intersection` that implements the ARM concept of `Edge_segment_cross_section` (see 4.2.311).

EXPRESS specification:

```
*)
ENTITY edge_segment_cross_section
  SUBTYPE OF (volume_shape_intersection);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
    SELF\representation_relationship ||
    SELF\shape_representation_relationship ||
    SELF\volume_shape_intersection ||
    SELF\edge_segment_cross_section))) = 0;
END_ENTITY; -- edge_segment_cross_section
(*
```

Formal propositions:

WR1: The `edge_segment_cross_section` shall not be a complex subtype of other than its direct super-types.

5.2.3.1.116 edge_segment_vertex

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An edge_segment_vertex is a physical_unit_datum that implements the ARM concept of Edge_segment_vertex (see 4.2.312).

EXPRESS specification:

```

*)
ENTITY edge_segment_vertex
  SUBTYPE OF (physical_unit_datum);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\physical_unit_datum ||
    SELF\edge_segment_vertex)) = 0;

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    pd\property_definition.description = 'shape element characterization'
  )) = 1;

  WR3: SELF\shape_aspect.of_shape\property_definition.definition.
    frame_of_reference\application_context_element.name
    IN ['physical design occurrence', 'physical design usage'];

  WR4: SELF\shape_aspect.name = 'single datum';
END_ENTITY; -- edge_segment_vertex
(

```

Formal propositions:

WR1: The edge_segment_vertex shall not participate in a complex instance of shape_aspect other than with the physical_unit_datum supertype.

WR2: The edge_segment_vertex shall be related to one representation with a name of 'shape element characterization'.

WR3: The edge_segment_vertex shall be in the 'physical design occurrence' or 'physical design usage' context.

WR4: The edge_segment_vertex shall be a single datum.

5.2.3.1.117 ee_specification

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An ee_specification is a document which implements the ARM concepts of:

- Assembly_technology_specification (see 4.2.55);
- Design_specification (see 4.2.280);
- Ee_specification (see 4.2.338);
- Fabrication_technology_specification (see 4.2.369);
- Interface_specification (see 4.2.472);
- Language_reference_manual (see 4.2.497);
- Lead_form_specification (see 4.2.510);
- Material_specification (see 4.2.537);
- Process_specification (see 4.2.740);
- Surface_finish_specification (see 4.2.880);
- Test_specification (see 4.2.898).

EXPRESS specification:

```

*)
ENTITY ee_specification
  SUBTYPE OF (document);
WHERE

  WR1: SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'document source')) +
    SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'document source')) >= 1;

```

```

WR2: SELF\document.kind\document_type.product_data_type IN
    ['assembly technology specification',
     'design specification', 'fabrication technology specification',
     'interface specification', 'language reference manual',
     'lead form specification', 'material specification',
     'reference document', 'source code',
     'process specification', 'surface finish specification',
     'test specification'];
END_ENTITY; -- ee_specification
(*)

```

Formal propositions:

WR1: The ee_specification shall have at least one organization or person_and_organization specified as its 'document source'.

WR2: The ee_specification shall be a assembly_technology_specification, design_specification, fabrication_technology_specification, interface_specification, language_reference_manual, lead_form_specification, material_specification, reference document, source code, process_specification, surface_finish_specification, or a test_specification.

5.2.3.1.118 electric_charge_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An electric_charge_measure_with_unit is a measure_with_unit that implements the ARM concept of Electric_charge_measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY electric_charge_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\electric_charge_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = coulomb;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- electric_charge_measure_with_unit
(*)

```

Formal propositions:

WR1: The electric_charge_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The electric_charge_measure_with_unit shall reference an si_unit with the name of coulomb.

WR3: The unit_component shall be an si_unit.

5.2.3.1.119 electrical_isolation_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An electrical_isolation_component_shape_aspect is a filled_area_material_removal_component_shape_aspect that implements the ARM concept of Electrical_isolation_laminate_component (see 4.2.342).

EXPRESS specification:

```
*)
ENTITY electrical_isolation_component_shape_aspect
  SUBTYPE OF (filled_area_material_removal_component_shape_aspect);
END_ENTITY; -- electrical_isolation_component_shape_aspect
(*
```

5.2.3.1.120 electrical_isolation_removal_template_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An electrical_isolation_removal_template_definition is a part_template_definition that implements the ARM concept of Electrical_isolation_removal_template (see 4.2.344) .

EXPRESS specification:

```
*)
ENTITY electrical_isolation_removal_template_definition
  SUBTYPE OF (part_template_definition);
(*
WHERE
```

```

WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' TEXT_LITERAL' IN
TYPEOF (it))) = 1)) = 1))) = 0);

WR2: (SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
[ ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MEASURE_REPRESENTATION_ITEM',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name =
' maximum font vertical extent')))) = 1
)) = 1))) = 0);

WR3: (SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
[ ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MEASURE_REPRESENTATION_ITEM',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name =
' maximum font horizontal extent')))) = 1
)) = 1))) = 0);
*)
END_ENTITY; -- electrical_isolation_removal_template_definition
( *

```

5.2.3.1.121 electrical_network

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `electrical_network` is a `functional_unit` that implements the ARM concept of `Electrical_network_-definition` (see 4.2.345).

EXPRESS specification:

```
*)
ENTITY electrical_network
  SUBTYPE OF (functional_unit);
END_ENTITY; -- electrical_network
(*
```

5.2.3.1.122 `electromotive_force_measure_with_unit`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `electromotive_force_measure_with_unit` is a `measure_with_unit` that implements the ARM concept of `Electromotive_force_measure_with_unit` (see ??).

EXPRESS specification:

```
*)
ENTITY electromotive_force_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: sizeof(typeof(self) - (typeof(self\measure_with_unit ||
    self\electromotive_force_measure_with_unit))) = 0;

  WR2: self\measure_with_unit.unit_component\si_unit.name = volt ;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    typeof(self\measure_with_unit.unit_component);
END_ENTITY; -- electromotive_force_measure_with_unit
(*
```

Formal propositions:

WR1: The `electromotive_force_measure_with_unit` shall not be any other type of `measure_with_unit`.

WR2: The `electromotive_force_measure_with_unit` shall reference an `si_unit` with a name of `volt`.

WR3: The `unit_component` shall be an `si_unit`.

5.2.3.1.123 `energy_measure_with_unit`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An energy_measure_with_unit is a measure_with_unit that implements the ARM concept of Energy_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY energy_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\energy_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = joule;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- energy_measure_with_unit
(*
```

Formal propositions:

WR1: The energy_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The energy_measure_with_unit shall reference an si_unit with a name of joule.

WR3: The unit_component shall be an si_unit.

5.2.3.1.124 evaluated_characteristic

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An evaluated_characteristic is a representation and a representation_relationship that implements the ARM concept of Evaluated_characteristic (see 4.2.356).

EXPRESS specification:

```

*)
ENTITY evaluated_characteristic
  SUBTYPE OF (representation, representation_relationship);
UNIQUE
  UR1: SELF\representation_relationship.rep_1, SELF\representation_relationship.rep_2;
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
    SELF\representation_relationship ||
    SELF\representation_relationship_with_transformation ||
    SELF\evaluated_characteristic))) = 0;

  WR2: SELF\representation_relationship.rep_1 <>
    SELF\representation_relationship.rep_2;

  WR3: SELF\representation_relationship.name = '';

  WR4: SELF\representation.name = '';

  WR5: SELF\representation_relationship.rep_1.name = 'planned characteristic';
END_ENTITY; -- evaluated_characteristic
(*

```

Formal propositions:

UR1: The combination of rep_1 and rep_2 shall be unique within a population of evaluated_characteristic.

WR1: The evaluated_characteristic shall only be a representation, representation_relationship, or representation_relationship_with_transformation in addition to itself.

WR2: The evaluated_characteristic rep_1 shall not be the rep_2.

WR3: The name attribute inherited from representation_relationship shall be an empty string.

WR4: The name attribute inherited from representation shall be an empty string.

WR5: The name attribute of representation_relationship.rep_1 shall be 'planned characteristic'.

5.2.3.1.125 evaluation_product_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An evaluation_product_definition is a product_definition that implements the ARM concept of: Evaluation_product_definition (see 4.2.357)

EXPRESS specification:

```

*)
ENTITY evaluation_product_definition
  SUBTYPE OF (product_definition);
END_ENTITY; -- evaluation_product_definition
( *

```

5.2.3.1.126 external_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An external_definition is a characterized_object and an externally_defined_representation_item and a descriptive_representation_item that implements the ARM concept of External_definition (see 4.2.361).

EXPRESS specification:

```

*)
ENTITY external_definition
  SUPERTYPE OF(ONEOF(bond_category))
  SUBTYPE OF (characterized_object, descriptive_representation_item,
              externally_defined_representation_item);
WHERE

  WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;
END_ENTITY; -- external_definition
( *

```

Formal propositions:

WR1: The external_definition shall be referenced by exactly one document.

5.2.3.1.127 externally_defined_assembly_definition

An externally_defined_assembly_definition is an externally_defined_physical_unit that implements the ARM concept of Pca_usage_view (see 4.2.641).

EXPRESS specification:

```

*)
ENTITY externally_defined_assembly_definition
  SUBTYPE OF (externally_defined_physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: NOT EXISTS(SELF\product_definition.name) OR
        (SELF\product_definition.name = 'assembly module');

  WR3: (NOT (SELF.frame_of_reference\application_context_element.name = 'physical design usage'
        OR (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL' IN
        TYPEOF (sa)) AND
        (sa\shape_aspect.description = 'pca terminal')))) >= 2))) = 0);
END_ENTITY; -- externally_defined_assembly_definition
(*

```

Formal propositions:

WR1: The name shall exist.

WR2: The externally_defined_assembly_definition shall be a definition of a product that has a name of 'assembly module'.

WR3: If the externally_defined_assembly_definition has a frame_of_reference name of 'physical design usage', it shall have at least two shape_aspects that are assembly_module_terminals with a description of 'pca terminal'.

5.2.3.1.128 externally_defined_bare_die

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An externally_defined_bare_die is an externally_defined_physical_unit that implements the ARM concept of Bare_die (see 4.2.59).

EXPRESS specification:

```

*)
ENTITY externally_defined_bare_die
  SUBTYPE OF (externally_defined_physical_unit);
WHERE

  WR1: SIZEOF (QUERY (ifdu <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' +
    'RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'implemented function') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' IN
    TYPEOF (ifdu.relying_product_definition)) AND
    (ifdu.relying_product_definition.frame_of_reference.name =
    'functional design usage')))) = 1;

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (dut <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'device unit technology') |
    dut.relying_property_definition\property_definition.name =
    'unit technology')) = 1)) = 1;

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE_TERMINAL' IN
    TYPEOF (sa))) >= 2))) = 0;
END_ENTITY; -- externally_defined_bare_die
(

```

Formal propositions:

WR1: The externally_defined_bare_die shall be related to exactly one functional_unit with a frame_of_reference name of 'functional design usage' as the 'implemented function'.

WR2: The externally_defined_bare_die shall have exactly one property_definition that is related to exactly one property_definition with a name of 'unit technology' as the 'device unit technology'.

WR3: The externally_defined_bare_die shall have at least two shape_aspects that are bare_die_terminals.

5.2.3.1.129 externally_defined_functional_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An externally_defined_functional_unit is a functional_unit and an externally_defined_product_definition that implements the ARM concepts of:

- Ee_product_definition (see 4.2.323);
- Externally_defined_ee_product_definition (see 4.2.364);
- Functional_unit_definition (see 4.2.401);
- Functional_unit_usage_view (see 4.2.410).

EXPRESS specification:

```

*)
ENTITY externally_defined_functional_unit
  SUBTYPE OF (functional_unit, externally_defined_product_definition);
END_ENTITY; -- externally_defined_functional_unit
( *
```

5.2.3.1.130 externally_defined_interconnect_definition

An externally_defined_interconnect_definition is an externally_defined_physical_unit that implements the ARM concept of Pcb_usage_view (see 4.2.643).

EXPRESS specification:

```

*)
ENTITY externally_defined_interconnect_definition
  SUBTYPE OF (externally_defined_physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: NOT EXISTS(SELF\product_definition.name) OR
        (SELF\product_definition.name = 'interconnect module');
END_ENTITY; -- externally_defined_interconnect_definition
( *
```

Formal propositions:

WR1: The name shall exist.

WR2: The externally_defined_interconnect_definition shall be a definition with a name of 'interconnect module'.

5.2.3.1.131 externally_defined_package

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An externally_defined_package is an externally_defined_physical_unit that implements the ARM concepts of Altered_package (see 4.2.8) and Package (see 4.2.573).

EXPRESS specification:

```

*)
ENTITY externally_defined_package
  SUBTYPE OF (externally_defined_physical_unit);
WHERE

  WR1: SELF.frame_of_reference.name = 'physical design usage';

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
      'registered case style')) >= 1))) = 0;

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr)) |
    sr_pdr.used_representation\representation.name = 'seating plane')) = 1))) = 0;

```

```

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY' IN
    TYPEOF (sa))) <= 1))) = 0;

WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_TERMINAL' IN
    TYPEOF (sa))) >= 1))) = 0;

WR6: (NOT (SELF\product_definition.description = 'altered package')) OR
  (SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    (pdr\product_definition_relationship.name = 'package preparation') AND
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PACKAGE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PACKAGE'] *
      TYPEOF (pdr.relateing_product_definition)) = 1))) = 1);
END_ENTITY; -- externally_defined_package
(*

```

Formal propositions:

WR1: The externally_defined_package shall have a frame_of_reference name of 'physical design usage'.

WR2: The externally_defined_package shall have at least one representation with the name of 'registered case style'.

WR3: The externally_defined_package shall have exactly one shape_representation with the name of 'seating plane'.

WR4: The externally_defined_package shall have at most one package_body.

WR5: The externally_defined_package shall have at least one package_terminal.

WR6: If the `externally_defined_package` has a description of 'altered package', it shall be related to exactly one package, `externally_defined_package`, or `library_defined_package` as the 'package preparation'.

Informal propositions:

IP1: Each three-dimensional shape representation of the `externally_defined_package` shall have exactly one `geometric_representation_item` that is an `axis2_placement_3d` and that `axis2_placement_3d` shall define the location and orientation of the seating plane in that representation. The normal of the seating plane in that representation shall be congruent with and the same direction as the Z axis of the `axis2_placement_3d`. The plane of the seating plane in that representation shall be congruent with the plane established by the X and Y axes of the `axis2_placement_3d`.

5.2.3.1.132 `externally_defined_packaged_connector`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `externally_defined_packaged_connector` is an `externally_defined_packaged_part` that implements the ARM concept of `Packaged_connector` (see 4.2.587).

EXPRESS specification:

```
*)
ENTITY externally_defined_packaged_connector
  SUBTYPE OF (externally_defined_packaged_part);
END_ENTITY; -- externally_defined_packaged_connector
(*
```

Informal propositions:

IP1: There shall be at least one `shape_aspect` that is a `packaged_connector_terminal_relationship` for the `externally_defined_packaged_connector`.

5.2.3.1.133 `externally_defined_packaged_part`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `externally_defined_packaged_part` is an `externally_defined_physical_unit` that implements the ARM concepts of `Altered_packaged_part` (see 4.2.10) and `Packaged_part` (see 4.2.591).

EXPRESS specification:

```

*)
ENTITY externally_defined_packaged_part
  SUPERTYPE OF (externally_defined_packaged_connector)
  SUBTYPE OF (externally_defined_physical_unit);
WHERE

  WR1: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART_TERMINAL' IN
      TYPEOF (sa))) >= 2))) = 0));

  WR2: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
    (SIZEOF (USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MATERIAL_DESIGNATION.DEFINITIONS')) = 1);

  WR3: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
    (SIZEOF (QUERY (ifu <* QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'implemented function') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT' IN
      TYPEOF (ifu.relatering_product_definition)) AND
    (ifu.relatering_product_definition.frame_of_reference.name =
      'functional design usage')))) = 1);

  WR4: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
    (SIZEOF (QUERY (upkg <* QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'used package') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE' IN
      TYPEOF (upkg.related_product_definition))) = 1);

```

```

WR5: (NOT (SELF\product_definition.description =
'altered packaged part')) OR
(SIZEOF (QUERY (bpp <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'base packaged part') |
(SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] *
TYPEOF (bpp.relatng_product_definition)) = 1) AND
(bpp.relatng_product_definition.frame_of_reference.name =
'physical design usage')) >= 1);

WR6: (NOT (SELF\product_definition.description =
'altered packaged part')) OR
(SIZEOF (QUERY (upkg <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'used package') |
SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] *
TYPEOF (upkg.relatng_product_definition)) = 1)) >= 1);
END_ENTITY; -- externally_defined_packaged_part
(*)

```

Formal propositions:

WR1: If the externally_defined_packaged_part has a frame_of_reference name of 'physical design usage', it shall have at least one shape_aspect that is a packaged_part_terminal.

WR2: If the externally_defined_packaged_part has a frame_of_reference name of 'physical design usage', it shall have exactly one material_designation.

WR3: If the externally_defined_packaged_part has a frame_of_reference name of 'physical design usage', it shall be related to exactly one functional_unit with a frame_of_reference name of 'functional design usage' as the 'implemented function'.

WR4: If the externally_defined_packaged_part has a frame_of_reference name of 'physical design usage', it shall relate to exactly one package to specify the 'used package'.

WR5: If the externally_defined_packaged_part has a description of 'altered packaged part', it shall be related to at least one packaged_part, externally_defined_packaged_part, or library_defined_packaged_part with a frame_of_reference name of 'physical design usage' as the 'base packaged part'.

WR6: If the `externally_defined_packaged_part` has a description of 'altered packaged part', it shall be related to at least one `package`, `externally_defined_package`, or `library_defined_package` as the 'used package'.

5.2.3.1.134 `externally_defined_physical_unit`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `externally_defined_physical_unit` is a `physical_unit` and an `externally_defined_product_definition` that implements the ARM concepts of:

- `Assembly_module_usage_view` (see 4.2.51);
- `Ee_product_definition` (see 4.2.323);
- `Externally_defined_ee_product_definition` (see 4.2.364);
- `Interconnect_module_usage_view` (see 4.2.465);
- `Physical_unit` (see 4.2.664);
- `Physical_unit_usage_view` (see 4.2.672).

EXPRESS specification:

```

*)
ENTITY externally_defined_physical_unit
  SUPERTYPE OF (ONEOF (externally_defined_package, externally_defined_packaged_part,
    externally_defined_bare_die, externally_defined_assembly_definition,
    externally_defined_interconnect_definition))
  SUBTYPE OF (physical_unit, externally_defined_product_definition);
END_ENTITY; -- externally_defined_physical_unit
( *
```

5.2.3.1.135 `externally_defined_product_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `externally_defined_product_definition` is a `product_definition` and an `externally_defined_item` that implements the ARM concept of: `Externally_defined_ee_product_definition` (see 4.2.364)

EXPRESS specification:

```

*)
ENTITY externally_defined_product_definition
  SUPERTYPE OF (library_defined_product_definition)
  SUBTYPE OF (product_definition, externally_defined_item);
END_ENTITY; -- externally_defined_product_definition
( *

```

5.2.3.1.136 externally_defined_representation_item

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An externally_defined_representation_item is an externally_defined_item and a representation_item that implements the ARM concept of External_definition (see 4.2.361).

EXPRESS specification:

```

*)
ENTITY externally_defined_representation_item
  SUBTYPE OF (externally_defined_item, representation_item);
WHERE

  WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;
END_ENTITY; -- externally_defined_representation_item
( *

```

Formal propositions:

WR1: The externally_defined_representation_item shall be referenced by exactly one applied_document_reference.

5.2.3.1.137 fabrication_joint

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A fabrication_joint is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Fabrication_joint (see 4.2.368).

EXPRESS specification:

```

*)
ENTITY fabrication_joint
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (SELF.related_shape_aspect)) AND
    (SELF.related_shape_aspect\shape_aspect.description IN
    ['via terminal', 'printed component join terminal',
    'non functional land join terminal',
    'land join terminal', 'conductive interconnect element terminal',
    'component termination passage join terminal']));

  WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (SELF.relate_shape_aspect)) AND
    (SELF.relate_shape_aspect\shape_aspect.description IN
    ['via terminal', 'printed component join terminal',
    'non functional land join terminal',
    'land join terminal', 'conductive interconnect element terminal',
    'component termination passage join terminal']));

  WR3: SIZEOF (QUERY (ajm <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'auxiliary joint material') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND' IN
    TYPEOF (ajm.related_shape_aspect)) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN
    TYPEOF (ajm.related_shape_aspect)) AND
    (ajm.related_shape_aspect\shape_aspect.description =
    'stratum feature template component')))) <= 1;

  WR4: acyclic_shape_aspect_relationship(SELF,
    [SELF\shape_aspect_relationship.related_shape_aspect],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FABRICATION_JOINT');
END_ENTITY; -- fabrication_joint
(

```

Formal propositions:

WR1: The fabrication_joint relating_shape_aspect attribute shall relate to exactly one component_terminals that have a description of 'via terminal', 'printed component join terminal', 'non functional land join terminal', 'land join terminal', 'conductive interconnect element terminal', or 'component termination passage join terminal' to specify the 'fabrication features'.

WR2: The fabrication_joint related_shape_aspect attribute shall relate to exactly one component_terminals that have a description of 'via terminal', 'printed component join terminal', 'non functional land join terminal', 'land join terminal', 'conductive interconnect element terminal', or 'component termination passage join terminal' to specify the 'fabrication features'.

WR3: The fabrication_joint shall relate to either at most one land or to at most one component_shape_aspect that has a description of 'stratum feature template component', to specify the 'auxiliary joint material'.

WR4: The fabrication_joint shall not participate in a cyclic relationship.

5.2.3.1.138 feature_shape_occurrence_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A feature_shape_occurrence_relationship is a representation and a shape_representation_relationship that implements the ARM concept of Feature_shape_occurrence_relationship (see ??).

EXPRESS specification:

```

*)
ENTITY feature_shape_occurrence_relationship
  SUBTYPE OF (representation, shape_representation_relationship);
UNIQUE
  UR1: SELF\representation_relationship.rep_1, SELF\representation_relationship.rep_2;
WHERE

  WR1: SELF\representation_relationship.rep_1 :<>: SELF\representation_relationship.rep_2;

  WR2: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
    SELF\representation_relationship ||
    SELF\shape_representation_relationship ||
    SELF\feature_shape_occurrence_relationship))) = 0;
END_ENTITY; -- feature_shape_occurrence_relationship
(

```

Formal propositions:

UR1: The combination of rep_1 and rep_2 shall be unique within a population of feature_shape_occurrence_relationship.

WR1: The feature_shape_occurrence_relationship rep_1 shall not be rep_2.

WR2: The feature_shape_occurrence_relationship shall not be a complex subtype of other than its direct supertypes.

5.2.3.1.139 fiducial

A fiducial is a component_shape_aspect that implements the ARM concept of Fiducial (see 4.2.373).

EXPRESS specification:

```

*)
ENTITY fiducial
  SUBTYPE OF (component_shape_aspect);
WHERE

  WR1: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
        AND (SELF.of_shape.definition\product_definition.description =
          'lamine component');
END_ENTITY; -- fiducial
( *
```

Formal propositions:

WR1: The fiducial shall be an aspect of a definition with a name of 'interconnect module'.

5.2.3.1.140 fiducial_part_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A fiducial_part_feature is a part_tooling_feature that implements the ARM concept of Fiducial_part_feature (see 4.2.374).

EXPRESS specification:

```
*)  
ENTITY fiducial_part_feature  
  SUBTYPE OF (part_tooling_feature);  
END_ENTITY; -- fiducial_part_feature  
(*
```

5.2.3.1.141 fiducial_stratum_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `fiducial_stratum_feature` is a `stratum_feature` that implements the ARM concept of `Fiducial_stratum_feature` (see 4.2.375).

EXPRESS specification:

```
*)  
ENTITY fiducial_stratum_feature  
  SUBTYPE OF (stratum_feature);  
END_ENTITY; -- fiducial_stratum_feature  
(*
```

5.2.3.1.142 filled_area_material_removal_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `filled_area_material_removal_component_shape_aspect` is a `material_removal_component_shape_aspect` that implements the ARM concept of `Filled_area_material_removal_laminate_component` (see 4.2.384).

EXPRESS specification:

```
*)  
ENTITY filled_area_material_removal_component_shape_aspect  
  SUBTYPE OF (material_removal_component_shape_aspect);  
END_ENTITY; -- filled_area_material_removal_component_shape_aspect  
(*
```

5.2.3.1.143 footprint_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `footprint_definition` is a `product_definition` that implements the ARM concepts of: `Footprint_definition` (see 4.2.388)

EXPRESS specification:

```

*)
ENTITY footprint_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF (SELF))) OR
    (SIZEOF (QUERY (docs <*
    SELF\product_definition_with_associated_documents.
    documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);

  WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) +
    SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ada.role\date_role.name = 'creation date')) = 1;

  WR3: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

  WR4: SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

  WR5: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

  WR6: SIZEOF (TYPEOF (SELF) - (TYPEOF (SELF\product_definition ||
    SELF\footprint_definition))) = 0;

  WR7: SELF.frame_of_reference.name IN ['layout design usage' ];

```

```

WR8: SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
    'PRODUCTS') |
    prpc\product_category.name = 'template model')) = 1;
END_ENTITY; -- footprint_definition
(*

```

Formal propositions:

WR1: If the footprint_definition is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR2: The footprint_definition shall have exactly one creation date.

WR3: The footprint_definition shall have exactly one approval.

WR4: The footprint_definition shall have at least one organization or person_and_organization as the creator.

WR5: The footprint_definition shall have exactly one security_classification.

WR6: The footprint_definition shall not be any other type of product_definition.

WR7: The footprint_definition shall have a frame_of_reference with a name of 'layout design usage'.

WR8: The footprint_definition shall be associated with a product that is a template model.

5.2.3.1.144 footprint_occurrence

A footprint_occurrence is an assembly_group_component_shape_aspect that implements the ARM concept of Footprint_occurrence (see 4.2.390).

EXPRESS specification:

```

*)
ENTITY footprint_occurrence
    SUBTYPE OF (assembly_group_component_shape_aspect);
END_ENTITY; -- footprint_occurrence
(*

```

5.2.3.1.145 footprint_occurrence_shape_aspect_relationship

A footprint_occurrence_shape_aspect_relationship is a shape_aspect_relationship that implements the ARM concept of Footprint_occurrence_sub_assembly_relationship (see 4.2.391)

EXPRESS specification:

```

*)
ENTITY footprint_occurrence_shape_aspect_relationship
  SUBTYPE OF (shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

  WR2: SELF\shape_aspect_relationship.name = 'footprint occurrence sub assembly relationship';
END_ENTITY; -- footprint_occurrence_shape_aspect_relationship
( *

```

Formal propositions:

WR1: The related shape_aspect shall not be the relating shape_aspect.

WR2: The footprint_occurrence_shape_aspect_relationship name shall be 'footprint occurrence sub assembly relationship'.

5.2.3.1.146 force_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A force_measure_with_unit is a measure_with_unit that implements the ARM concept of Force_measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY force_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
        SELF\force_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = newton;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
        TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- force_measure_with_unit
( *

```

Formal propositions:

WR1: The force_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The force_measure_with_unit shall reference an si_unit with a name of newton.

WR3: The unit_component shall be an si_unit.

5.2.3.1.147 frequency_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A frequency_measure_with_unit is a measure_with_unit that implements the ARM concept of Frequency_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY frequency_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\frequency_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = hertz;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- frequency_measure_with_unit
(*
```

Formal propositions:

WR1: The frequency_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The frequency_measure_with_unit shall reference an si_unit with a name of hertz.

WR3: The unit_component shall be an si_unit.

5.2.3.1.148 frozen_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An frozen_assignment is an approval_assignment that specifies a set of frozen_assigned_item. An item that is 'approved' is frozen. An item that is 'not yet approved' is not frozen.

EXAMPLE 1 A component placement is considered frozen (i.e., the component cannot be moved) if its next_assembly_usage_occurrence is a member of the set of frozen_assigned_item and the approval_status is "approved".

EXPRESS specification:

```
*)
ENTITY frozen_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF frozen_assigned_item;
WHERE

  WR1: SELF\approval_assignment.assigned_approval.status.name in ['approved', 'not yet approved'];
END_ENTITY; -- frozen_assignment
(*
```

Formal propositions:

WR1: The approval status shall either be 'approved' or 'not yet approved'.

5.2.3.1.149 functional_specification

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An functional_specification is a representation that implements the ARM concept of Functional_specification (see 4.2.396).

EXPRESS specification:

```
*)
ENTITY functional_specification
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    (rr\representation_relationship.name = 'functional characteristic category') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNAL_DEFINITION' IN
    TYPEOF (rr.rep_1.items[1])))) = 1;
```

```

WR2: sizeof (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') |
    (rr\representation_relationship.name = 'characterizing signal' +
    'for functional specification')
    )) = 1;

WR3: sizeof (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') |
    (rr\representation_relationship.name = 'reference signal for' +
    'functional specification')
    )) = 1;

WR4: sizeof (typeof (SELF) - typeof (SELF\representation ||
    SELF\functional_specification)) = 0;

WR5: SELF\representation.name = '';

WR6: sizeof (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_FUNCTIONAL_UNIT' IN typeof (pdr.definition.definition))
    )) = 0;
END_ENTITY; -- functional_specification
(*

```

Formal propositions:

- WR1:** The functional_specification shall be in exactly one 'functional characteristic category'.
- WR2:** The functional_specification shall reference exactly one 'characterizing signal for functional specification'.
- WR3:** The functional_specification shall reference exactly one 'reference signal for functional specification'.
- WR4:** The functional_specification shall be no other type of representation.
- WR5:** The functional_specification name shall be blank.
- WR6:** The functional_specification shall reference exactly one component_functional_unit.

5.2.3.1.150 functional_specification_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A functional_specification_definition is a functional_unit that implements the ARM concept of functional_specification_definition (see 4.2.397).

EXPRESS specification:

```

*)
ENTITY functional_specification_definition
  SUBTYPE OF (functional_unit);
WHERE

  WR1: SELF.frame_of_reference\application_context_element.name =
        'functional network design';

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_SPECIFICATION' IN TYPEOF (pdr.used_representation))
    )) > 0))) > 0;
END_ENTITY; -- functional_specification_definition
( *

```

Formal propositions:

WR1: The functional_specification_definition context shall be 'functional network design'.

WR2: The functional_specification_definition shall be related to at least one functional_specification.

5.2.3.1.151 functional_terminal_group

A functional_terminal_group is a group that specifies a type for the terminals assigned to the group.

EXPRESS specification:

```

*)
ENTITY functional_terminal_group
  SUBTYPE OF (group);
UNIQUE
  UR1: SELF\group.name;
END_ENTITY; -- functional_terminal_group
( *

```

Formal propositions:

UR1: The name shall be unique.

5.2.3.1.152 functional_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A functional_unit is the characterization of the functionality of an occurrence of an electrical device or of an abstract component. A functional_unit implements the ARM concepts of:

- Ee_product_definition (see 4.2.323);
- Electrical_network_definition (see 4.2.345);
- Functional_specification_definition (see 4.2.397);
- Functional_unit_definition (see 4.2.401);
- Functional_unit_network_definition (see 4.2.402);
- Functional_unit_usage_view (see 4.2.410);
- Thermal_network_definition (see 4.2.906).

EXPRESS specification:

```

*)
ENTITY functional_unit
  SUPERTYPE OF ((electrical_network)
                ANDOR (thermal_network)
                ANDOR (functional_specification_definition))
  SUBTYPE OF (product_definition);
WHERE

  WR1: SELF.frame_of_reference.name IN ['functional design usage',
    'functional network design'];

  WR2: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF (SELF))) OR
    (SIZEOF (QUERY (docs < *
    SELF\product_definition_with_associated_documents.documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);

  WR3: SIZEOF (QUERY (adta < * USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) = 1;

```



```

WR4: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

WR5: sizeof (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    sizeof (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

WR6: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

WR7: (NOT (SELF.frame_of_reference.name =
    'functional network design')) OR
    (sizeof (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' +
    'RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'design usage') |
    du.relatng_product_definition.frame_of_reference.name =
    'functional design usage')) = 1);

WR8: (NOT (sizeof([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'THERMAL_NETWORK',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELECTRICAL_NETWORK'] * TYPEOF(SELF)) = 1))
    OR (SELF.frame_of_reference.name = 'functional network design');
END_ENTITY; -- functional_unit
(*

```

Formal propositions:

WR1: The functional_unit shall have a frame_of_reference with a name of 'functional design usage' or 'functional network design'.

WR2: If the functional_unit is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR3: The functional_unit shall have exactly one creation date and time.

WR4: The functional_unit shall have exactly one approval.

WR5: The functional_unit shall have at least one organization or person_and_organization as the creator.

WR6: The functional_unit shall have exactly one security_classification.

WR7: If the functional_unit is a functional network design it shall be related to exactly one functional_unit that is a functional design usage as the design usage.

WR8: If the functional_unit is a thermal_network or an electrical_network the frame_of_reference attribute shall be satisfied by an entity with a name equal to 'functional network design'.

5.2.3.1.153 functional_unit_terminal_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A functional_unit_terminal_definition is a shape_aspect that implements the ARM concepts of Functional_unit_usage_view_terminal_definition (see 4.2.411) and Scalar_terminal_definition (see 4.2.799).

EXPRESS specification:

```

*)
ENTITY functional_unit_terminal_definition
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT' IN
        TYPEOF (SELF.of_shape.definition)) AND
        (SELF.of_shape.definition\product_definition.frame_of_reference.name =
        'functional design usage');

  WR2: SIZEOF (QUERY (pd2 <* QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION' IN
        TYPEOF (pd)) |
        SIZEOF (QUERY (funtdba <* QUERY (pdr <* USEDIN (pd2,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
        pdr\property_definition_relationship.name =
        'functional unit network terminal definition bus assignment') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION' IN
        TYPEOF
        (funtdba.relatng_property_definition.definition))) <= 1)) <= 1;

```

```

WR3: SIZEOF (QUERY (pd2 <* QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION' IN
TYPEOF (pd)) |
SIZEOF (QUERY (funtdna <* QUERY (pdr <* USEDIN (pd2,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name =
' functional unit network terminal definition node assignment') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' NETWORK_NODE_DEFINITION' IN
TYPEOF
(funtdna.relatng_property_definition.definition))) <= 1)) <= 1;
END_ENTITY; -- functional_unit_terminal_definition
( *

```

Formal propositions:

WR1: The functional_unit_terminal_definition shall be an aspect of the shape of a functional_unit that has a context element with a name of 'functional design usage'.

WR2: The functional_unit_terminal_definition shall have no more than one property definition that is related to at most one property of a bus_structural_definition as the 'functional unit network terminal definition bus assignment'.

WR3: The functional_unit_terminal_definition shall have no more than one property definition that is related to at most one property of a network_node_definition as the 'functional unit network terminal definition node assignment'.

5.2.3.1.154 geometric_tolerance_group

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A geometric_tolerance_group is a property_definition that implements the ARM concepts of Separate_geometric_tolerance_requirement (see 4.2.805) and Simultaneous_geometric_tolerance_requirement (see 4.2.826).

EXPRESS specification:

```

*)
ENTITY geometric_tolerance_group
  SUBTYPE OF (property_definition);
WHERE

```

```

WR1: SELF\property_definition.description
    IN ['separate requirement', 'simultaneous requirement'];

WR2: (NOT (SELF\property_definition.description =
    'separate requirement'))
    OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM'
    IN TYPEOF(pdr.related_property_definition)) AND
    (pdr\property_definition_relationship.name = 'group geometric tolerance')
    )) >=1);

WR3: (NOT (SELF\property_definition.description =
    'simultaneous requirement'))
    OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE'
    IN TYPEOF(pdr.related_property_definition)) AND
    (pdr\property_definition_relationship.name = 'group geometric tolerance'))
    )) >=2);

WR4: (NOT (SELF\property_definition.description =
    'separate requirement'))
    OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINEAR_PROFILE_TOLERANCE' IN
    TYPEOF(pdr.related_property_definition)) AND
    (pdr.related_property_definition\property_definition.name =
    'linear profile refinement') AND
    (pdr\property_definition_relationship.name = 'group geometric tolerance') )
    )) =0);

WR5: (NOT (SELF\property_definition.description =
    'separate requirement'))
    OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE_PROFILE_TOLERANCE' IN
    TYPEOF(pdr.related_property_definition)) AND
    (pdr.related_property_definition\property_definition.name =
    'surface profile refinement') AND
    (pdr\property_definition_relationship.name = 'group geometric tolerance') )
    )) =0);

```

```

WR6: (NOT (SELF\property_definition.description =
'separate requirement'))
OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITION_TOLERANCE' IN
TYPEOF(pdr.related_property_definition)) AND
(pdr.related_property_definition\property_definition.name =
'feature relating position') AND
(pdr\property_definition_relationship.name = 'group geometric tolerance') )
)) =0);

WR7: (NOT (SELF\property_definition.description =
'separate requirement'))
OR (SIZEOF (QUERY(pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
(pdr\property_definition_relationship.name = 'group geometric tolerance') AND
(SIZEOF(['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINEAR_PROFILE_TOLERANCE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITION_TOLERANCE'] * TYPEOF(pdr.related_property_definition)) > 0)
)) =0);
END_ENTITY; -- geometric_tolerance_group
(*

```

Formal propositions:

WR1: The `geometric_tolerance_group` shall have a description of either 'separate requirement' or 'simultaneous requirement'.

WR2: If the `geometric_tolerance_group` has a description of 'separate requirement', the `geometric_tolerance_group` shall relate to at least one `geometric_tolerance_with_specified_datum_system` to specify the 'group geometric tolerance'.

WR3: If the `geometric_tolerance_group` has a description of 'simultaneous requirement', the `geometric_tolerance_group` shall relate to at least two `physical_unit_geometric_tolerances` to specify the 'group geometric tolerance'.

WR4: If the `geometric_tolerance_group` has a description of 'separate requirement', the `geometric_tolerance_group` shall relate to exactly zero `linear_profile_tolerance` with a name of 'linear profile refinement' to specify the 'group geometric tolerance'.

WR5: If the `geometric_tolerance_group` has a description of 'separate requirement', the `geometric_tolerance_group` shall relate to exactly zero `surface_profile_tolerance` with a name of 'surface profile refinement' to specify the 'group geometric tolerance'.

WR6: If the `geometric_tolerance_group` has a description of 'separate requirement', the `geometric_tolerance_group` shall relate to exactly zero `position_tolerance` with a name of 'feature relating position' to specify the 'group geometric tolerance'.

WR7: If the `geometric_tolerance_group` has a description of 'separate requirement', the `geometric_tolerance_group` shall relate to exactly zero `physical_unit_geometric_tolerance`, to specify the 'group geometric tolerance', that are not either `linear_profile_tolerance`, `surface_profile_tolerance`, or `position_tolerance`.

Informal propositions:

IP1: If the `geometric_tolerance_group` has a description of 'separate requirement' and the `geometric_tolerance_group` relates to at least two `geometric_tolerance_with_specified_datum_systems` to specify the 'group geometric tolerance', all of the `geometric_tolerance_with_specified_datum_system` shall relate to the same `property_definition` to specify the 'referenced datum system definition', and that `property_definition` shall have a description of either 'datum system property with material conditions' or 'datum system property without material conditions'.

IP2: If the `geometric_tolerance_group` has a description of 'simultaneous requirement' and the `geometric_tolerance_group` relates to at least two `physical_unit_geometric_tolerances` to specify the 'group geometric tolerance', all of the `physical_unit_geometric_tolerance` either shall not relate to a `property_definition` to specify the 'referenced datum system definition' or shall relate to the same `property_definition` to specify the 'referenced datum system definition', and that `property_definition` shall have a description of either 'datum system property with material conditions' or 'datum system property without material conditions'.

IP3: If the `geometric_tolerance_group` has a description of 'simultaneous requirement', the `geometric_tolerance_group` shall relate to exactly zero `physical_unit_geometric_tolerance`, to specify the 'group geometric tolerance', that are not either a `linear_profile_tolerance` with a description of 'linear profile refinement', a `surface_profile_tolerance` with a description of 'surface profile refinement', or a `position_tolerance` with a description of 'feature relating position'.

5.2.3.1.155 `geometric_tolerance_with_specified_datum_system`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `geometric_tolerance_with_specified_datum_system` is a `physical_unit_geometric_tolerance` that implements the ARM concepts of:

- `Boundary_based_position_tolerance` (see 4.2.77);
- `Composite_feature_locating_position_tolerance` (see 4.2.157);
- `Composite_linear_profile_locating_tolerance` (see 4.2.160);
- `Composite_surface_profile_locating_tolerance` (see 4.2.165);

- Geometric_tolerance_with_referenced_datum_system (see 4.2.422);
- Position_tolerance_with_referenced_datum_system (see 4.2.696);
- Profile_of_any_line_tolerance_with_referenced_datum_system (see 4.2.745);
- Profile_of_any_surface_tolerance_with_referenced_datum_system (see 4.2.748).

EXPRESS specification:

*)

ENTITY geometric_tolerance_with_specified_datum_system

SUBTYPE OF (physical_unit_geometric_tolerance);

WHERE

```
WR1: SIZEOF([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'ANGULARITY_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'POSITION_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'CIRCULAR_RUNOUT_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'LINEAR_PROFILE_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'SURFACE_PROFILE_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'CONCENTRICITY_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PARALLELISM_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PERPENDICULARITY_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'SYMMETRY_TOLERANCE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'TOTAL_RUNOUT_TOLERANCE'] * TYPEOF(SELF)) = 1;
```

```
WR2: SIZEOF (QUERY(pdr <* USEDIN(SELF,
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
              ((pdr.related_property_definition\property_definition.description =
                'datum system property with material conditions')
               OR (pdr.related_property_definition\property_definition.description =
                'datum system property without material conditions'))
              AND (pdr\property_definition_relationship.name = 'referenced datum system definiti
              )) = 1;
```

```

WR3: SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
     + 'GEOMETRIC_TOLERANCE_GROUP' IN
      TYPEOF(pdr.related_property_definition))
    AND (pdr.related_property_definition\property_definition.description =
        'separate requirement')
    AND (pdr\property_definition_relationship.name = 'group geometric tolerance')
    )) <= 1;
END_ENTITY; -- geometric_tolerance_with_specified_datum_system
(*

```

Formal propositions:

WR1: The `geometric_tolerance_with_specified_datum_system` shall be one of the following types: `angularity_tolerance`, `position_tolerance`, `circular_runout_tolerance`, `linear_profile_tolerance`, `surface_profile_tolerance`, `concentricity_tolerance`, `parallelism_tolerance`, `perpendicularity_tolerance`, `symmetry_tolerance`, or `total_runout_tolerance`.

WR2: The `geometric_tolerance_with_specified_datum_system` shall relate to exactly one `property_definition` that has the description of either 'datum system property with material conditions' or 'datum system property without material conditions' to specify the 'referenced datum system definition'.

WR3: The `geometric_tolerance_with_specified_datum_system` shall be related to at most one `geometric_tolerance_group` with a description of 'separate requirement' as the 'group geometric tolerance'.

5.2.3.1.156 group_product_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `group_product_definition` is a `component_definition` that implements the ARM concept of `Placement_group` (see 4.2.674).

EXPRESS specification:

```

*)
ENTITY group_product_definition
    SUBTYPE OF (component_definition);
WHERE

    WR1: SELF.frame_of_reference.name = 'design requirement';

```



```

WR2: (NOT (SELF\product_definition.description =
                                'placement group')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REQUIREMENTS_PROPERTY' IN
      TYPEOF (pd))) >= 1);
END_ENTITY; -- group_product_definition
(*

```

Formal propositions:

WR1: The group_product_definition shall have a frame_of_reference name of 'design requirement'.

WR2: If the group_product_definition has a description of 'placement group', it shall have at least one requirements_property.

5.2.3.1.157 group_shape_aspect

A group_shape_aspect is a shape_aspect that implements the ARM concepts of:

- Interconnect_module_constraint_region (see 4.2.446);
- Placement_group (see 4.2.674);
- Termination_constraint (see 4.2.892).

EXPRESS specification:

```

*)
ENTITY group_shape_aspect
  SUBTYPE OF (shape_aspect);
WHERE

```

```

WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION' IN
      TYPEOF (SELF\shape_aspect.of_shape.definition);

```

```

WR2: NOT(SELF\shape_aspect.description IN
      ['interconnect module constraint region']) OR
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PHYSICAL_UNIT' IN
       TYPEOF (SELF\shape_aspect.of_shape.definition)) AND
      (EXISTS(SELF\shape_aspect.of_shape.definition\product_definition.name)) AND
      (NOT EXISTS(SELF\shape_aspect.of_shape.definition\product_definition.name) OR
      (SELF\shape_aspect.of_shape.definition\product_definition.name = 'interconnect modu

```

```

WR3: ((NOT(SELF\shape_aspect.description IN
          ['interconnect module constraint region'])) OR
      (SIZEOF( QUERY ( pd <* USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION')
          | (SIZEOF( QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
          | ((pdr\property_definition_relationship.name = 'constrained ob
AND (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY') IN
      TYPEOF(pdr.related_property_definition.definition)))
      )) = 1)
    )) = 1));

```

```

WR4: (NOT(SELF\shape_aspect.description IN
          ['interconnect module constraint region'])) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      SIZEOF (QUERY (it <* pdr.used_representation.items |
          ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'DESCRIPTIVE_REPRESENTATION_ITEM' IN
          TYPEOF (it)) AND
          (it\representation_item.name = 'design specific purpose')))) = 1)) = 1))) = 0);

```

```

WR5: (NOT(SELF\shape_aspect.description IN
['interconnect module constraint region'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
(it\representation_item.name = 'keepout') AND
(it\descriptive_representation_item.description IN ['true', 'false']))) = 1)) = 1))

WR6: (NOT(SELF\shape_aspect.description IN
['interconnect module constraint region'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation))) = 1)) = 1);

WR7: (NOT(SELF\shape_aspect.description IN
['interconnect module constraint region'])) OR
(SIZEOF( QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION')
| (SIZEOF( QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
| ((pdr\property_definition_relationship.name = 'requirement')
AND (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY') IN
TYPEOF(pdr.relatng_property_definition)))
)) = 1)
)) = 1);

WR8: (NOT(SELF\shape_aspect.description =
'termination constraint')) OR
(SIZEOF (QUERY (ctm <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATNG_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'constrained termination member') |
ctm.related_shape_aspect\shape_aspect.description =
'mating connector termination')) >= 2);

```

```

WR9: EXISTS(SELF\shape_aspect.description);

WR10: (NOT(SELF\shape_aspect.description IN
  ['placement group'])) OR
  (SIZEOF( QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')
    | (SIZEOF( QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | ((pdr\property_definition_relationship.name = 'requirement')
  AND (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY') IN
    TYPEOF(pdr.relatng_property_definition))
  AND
  (pdr.relatng_property_definition\property_definition.name = 'requirement'))
    ) = 1)
  )) = 1);

WR11: (NOT(SELF\shape_aspect.description IN
  ['termination constraint'])) OR
  (SIZEOF( QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')
    | (SIZEOF( QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | ((pdr\property_definition_relationship.name = 'termination usage constraint')
  AND (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY') IN
    TYPEOF(pdr.relatng_property_definition))
  AND
  (pdr.relatng_property_definition\property_definition.name = 'termination usage constraint')
    ) = 1)
  )) = 1);
END_ENTITY; -- group_shape_aspect
(*

```

Formal propositions:

WR1: The group_shape_aspect shall be an aspect of the shape of a product_definition.

WR2: If the group_shape_aspect has a description of 'interconnect module constraint region', it shall be an aspect of the shape of a 'interconnect module'.

WR3: If the group_shape_aspect has a description of 'interconnect module constraint region', it shall relate to exactly one interconnect_module_design_object_category to specify the 'constrained object'.

WR4: If the group_shape_aspect has a description of 'interconnect module constraint region', it shall have exactly one representation that contains exactly one descriptive_representation_item with a name of 'design specific purpose'.

WR5: If the group_shape_aspect has a description of 'interconnect module constraint region', it shall have exactly one representation that contains exactly one descriptive_representation_item with a name of 'keepout' and a description of either 'true' or 'false'.

WR6: If the group_shape_aspect has a description of 'interconnect module constraint region', it shall have exactly one property that has exactly one shape_representation.

WR7: If the group_shape_aspect has a description of 'interconnect module constraint region', or 'termination constraint', or 'placement group' it shall have exactly one requirements_property.

WR8: If the group_shape_aspect has a description of 'termination constraint', it shall relate to at least two shape_aspects with a description of 'mating connector termination' to specify a 'constrained termination member'.

WR9: The attribute description inherited from shape_aspect shall exist.

WR10: If the group_shape_aspect has a description of 'placement group' it shall have exactly one requirements_property.

WR11: If the group_shape_aspect has a description of 'termination constraint' it shall have exactly one requirements_property.

5.2.3.1.158 grouped_requirements_property

A grouped_requirements_property is a group and a requirements_property that implements the ARM concepts of Item_restricted_requirement_occurrence (see 4.2.478) and Layout_spacing_requirement_occurrence (see 4.2.509).

EXPRESS specification:

```
*)
ENTITY grouped_requirements_property
  SUBTYPE OF (group, requirements_property);
WHERE

  WR1: (NOT (SELF\group.name =
    'item restricted requirements property')) OR
    (SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT' IN
    TYPEOF (ga)) |
    SIZEOF (aga.items) = 1)) = 1));
```

```

WR2: (NOT (SELF\group.name =
    'layout spacing requirements property')) OR
(SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT' IN
    TYPEOF (ga)) |
    (SIZEOF (aga.items) = 2) AND
    (SIZEOF (QUERY (rp <* QUERY (it <* aga.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESIGN_OBJECT' IN
    TYPEOF (it)) |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESIGN_OBJECT' IN
    TYPEOF (rp)) AND
    (rp\characterized_object.name IN [
    'assembly module assembly component',
    'assembly module component terminal',
    'assembly module component',
    'assembly module macro component join terminal',
    'assembly module macro component',
    'assembly module',
    'bare die component terminal',
    'bare die component',
    'bonded conductive base blind via',
    'buried via',
    'cable component',
    'component feature',
    'component mounting feature',
    'component termination passage and contact size dependent land',
    'component termination passage and contact size dependent non functional land',
    'component termination passage dependent land',
    'component termination passage dependent non functional land',
    'component termination passage interface terminal',
    'component termination passage join terminal',
    'component termination passage',
    'conductive interconnect element terminal',
    'conductive interconnect element with pre defined transitions',
    'conductive interconnect element with user defined single transition',
    'connected area component',
    'contact size dependent land',
    'contact size dependent non functional land',
    'cutout edge segment',
    'cutout',
    'dielectric material passage',
    'electrical isolation laminate component',
    'embedded component terminal',
    'fiducial',
    'fill area',
    'filled area material removal laminate component',
    'inter stratum feature',

```

'interconnect component interface terminal',
 'interconnect component join terminal',
 'interconnect module assembly component',
 'interconnect module component stratum based terminal',
 'interconnect module component surface feature',
 'interconnect module component terminal',
 'interconnect module component',
 'interconnect module edge segment',
 'interconnect module edge',
 'interconnect module macro component join terminal',
 'interconnect module macro component',
 'interface access material removal laminate component',
 'interface access stratum feature template component',
 'interface component',
 'interfacial connection',
 'internal probe access area',
 'join two physical connectivity definition supporting inter stratum feature',
 'join two physical connectivity definition supporting printed component',
 'laminate component',
 'laminate text component',
 'laminate text string component',
 'land interface terminal',
 'land join terminal',
 'land',
 'material removal laminate component',
 'minimally defined component terminal',
 'movable packaged component join terminal',
 'multi layer material removal laminate component',
 'multi layer stratum feature template component',
 'non conductive base blind via',
 'non functional land interface terminal',
 'non functional land join terminal',
 'non functional land',
 'packaged component join terminal',
 'packaged component',
 'packaged connector component interface terminal',
 'packaged connector component',
 'partially plated cutout',
 'partially plated interconnect module edge',
 'physical component',
 'physical connectivity interrupting cutout',
 'physical laminate component',
 'physical network supporting inter stratum feature',
 'plated conductive base blind via',
 'plated cutout edge segment',
 'plated cutout',
 'plated interconnect module edge segment',
 'plated interconnect module edge',
 'plated passage or unsupported passage',
 'plated passage',
 'primary stratum indicator symbol',
 'printed component join terminal',

```

'printed connector component interface terminal',
'probe access area',
'routed interconnect component',
'routed physical component',
'special symbol laminate component',
'stratum feature template component',
'stratum feature',
'thermal isolation laminate component',
'unrouted conductive interconnect element',
'unsupported passage dependent non functional land',
'unsupported passage',
'via and contact size dependent land',
'via and contact size dependent non functional land',
'via dependent land',
'via dependent non functional land',
'via terminal',
'via']))) = 2))) = 1));

```

```

WR3: (NOT (SELF\group.name =
'layout spacing requirements property')) OR
(SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_GROUP_ASSIGNMENT' IN
TYPEOF (ga)) |
(SIZEOF (QUERY (rp <* QUERY (it <* aga.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESIGN_OBJECT' IN
TYPEOF (it)) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESIGN_OBJECT' IN
TYPEOF (rp)) AND
(rp\characterized_object.description =
'dependent design object category')))) = 1))) = 1));

```



```

WR4: (NOT (SELF\group.name =
      'layout spacing requirements property')) OR
      (SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT' IN
      TYPEOF (ga)) |
      (SIZEOF (QUERY (rp <* QUERY (it <* aga.items |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESIGN_OBJECT' IN
      TYPEOF (it)) |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESIGN_OBJECT' IN
      TYPEOF (rp)) AND
      (rp\characterized_object.description =
      'reference design object category')))) = 1))) = 1);
END_ENTITY; -- grouped_requirements_property
(*

```

Formal propositions:

WR1: If the grouped_requirements_property has a group name of 'item restricted requirements property', it shall be assigned to exactly one item.

WR2: If the `grouped_requirements_property` has a group name of 'layout spacing requirements property', it shall be assigned to exactly two **design_objects** with a name of 'assembly module assembly component', 'assembly module component terminal', 'assembly module component', 'assembly module macro component join terminal', 'assembly module macro component', 'assembly module', 'bare die component terminal', 'bare die component', 'bonded conductive base blind via', 'buried via', 'cable component', 'component feature', 'component mounting feature', 'component termination passage and contact size dependent land', 'component termination passage and contact size dependent non functional land', 'component termination passage dependent land', 'component termination passage dependent non functional land', 'component termination passage interface terminal', 'component termination passage join terminal', 'component termination passage', 'conductive interconnect element terminal', 'conductive interconnect element with pre defined transitions', 'conductive interconnect element with user defined single transition', 'connected area component', 'contact size dependent land', 'contact size dependent non functional land', 'cutout edge segment', 'cutout', 'dielectric material passage', 'electrical isolation laminate component', 'embedded component terminal', 'fiducial', 'fill area', 'filled area material removal laminate component', 'inter stratum feature', 'interconnect component interface terminal', 'interconnect component join terminal', 'interconnect module assembly component', 'interconnect module component stratum based terminal', 'interconnect module component surface feature', 'interconnect module component terminal', 'interconnect module component', 'interconnect module edge segment', 'interconnect module edge', 'interconnect module macro component join terminal', 'interconnect module macro component', 'interface access material removal laminate component', 'interface access stratum feature template component', 'interface component', 'interfacial connection', 'internal probe access area', 'join two physical connectivity definition supporting inter stratum feature', 'join two physical connectivity definition supporting printed component', 'laminate component', 'laminate text component', 'laminate text string component', 'land interface terminal', 'land join terminal', 'land', 'material removal laminate component', 'minimally defined component terminal', 'movable packaged component join terminal', 'multi layer material removal laminate component', 'multi layer stratum feature template component', 'non conductive base blind via', 'non functional land interface terminal', 'non functional land join terminal', 'non functional land', 'packaged component join terminal', 'packaged component', 'packaged connector component interface terminal', 'packaged connector component', 'partially plated cutout', 'partially plated interconnect module edge', 'physical component', 'physical connectivity interrupting cutout', 'physical laminate component', 'physical network supporting inter stratum feature', 'plated conductive base blind via', 'plated cutout edge segment', 'plated cutout', 'plated interconnect module edge segment', 'plated interconnect module edge', 'plated passage or unsupported passage', 'plated passage', 'primary stratum indicator symbol', 'printed component join terminal', 'printed connector component interface terminal', 'probe access area', 'routed interconnect component', 'routed physical component', 'special symbol laminate component', 'stratum feature template component', 'stratum feature', 'thermal isolation laminate component', 'unrouted conductive interconnect element', 'unsupported passage dependent non functional land', 'unsupported passage', 'via and contact size dependent land', 'via and contact size dependent non functional land', 'via dependent land', 'via dependent non functional land', 'via terminal' or 'via'.

WR3: If the `grouped_requirements_property` has a group name of 'layout spacing requirements property', it shall be assigned to exactly one **design_object** with a description of 'dependent design object category'.

WR4: If the grouped_requirements_property has a group name of 'layout spacing requirements property', it shall be assigned to exactly one design_object with a description of 'reference design object category'.

5.2.3.1.159 guided_wave_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A guided_wave_terminal is a package_terminal that implements the ARM concept of Guided_wave_terminal (see 4.2.427).

EXPRESS specification:

```
*)
ENTITY guided_wave_terminal
  SUBTYPE OF (package_terminal);
END_ENTITY; -- guided_wave_terminal
(*
```

5.2.3.1.160 illuminance_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An illuminance_measure_with_unit is a measure_with_unit that implements the ARM concept of Illuminance_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY illuminance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\illuminance_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = lux;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- illuminance_measure_with_unit
(*
```

Formal propositions:

WR1: The illuminance_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The illuminance_measure_with_unit shall reference an si_unit with a name of lux.

WR3: The unit_component shall be an si_unit.

5.2.3.1.161 inductance_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An inductance_measure_with_unit is a measure_with_unit that implements the ARM concept of Inductance_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY inductance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\inductance_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = henry;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- inductance_measure_with_unit
(*
```

Formal propositions:

WR1: The inductance_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The inductance_measure_with_unit shall reference an si_unit with a name of henry.

WR3: The unit_component shall be an si_unit.

5.2.3.1.162 integral_shield

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An integral_shield is a component_shape_aspect that implements the ARM concept of Shield (see 4.2.821).

EXPRESS specification:

```

*)
ENTITY integral_shield
  SUBTYPE OF (component_shape_aspect);
WHERE

  WR1: SIZEOF (QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN TYPEOF(pdr.relying_property_definition))
    OR ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN TYPEOF(pdr.relying_property_definition)))
    AND (pdr\property_definition_relationship.name = 'shielded item')) >=1;
END_ENTITY; -- integral_shield
( *

```

Formal propositions:

WR1: The `integral_shield` shall be related to at least one `component_definition` or to at least one `component_shape_aspect` as the 'shielded item'.

5.2.3.1.163 inter_stratum_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `inter_stratum_feature` is a `component_shape_aspect` that implements the ARM concepts of:

- Cutout (see 4.2.208);
- Cutout_edge_segment (see 4.2.209);
- Dielectric_material_passage (see 4.2.289);
- Inter_stratum_feature (see 4.2.432);
- Interconnect_module_edge (see 4.2.449);
- Interconnect_module_edge_segment (see 4.2.450);
- Partially_plated_cutout (see 4.2.634);
- Partially_plated_interconnect_module_edge (see 4.2.635);
- Physical_connectivity_interrupting_cutout (see 4.2.655);

— Unsupported_passage (see 4.2.939).

EXPRESS specification:

*)

```
ENTITY inter_stratum_feature
  SUBTYPE OF (component_shape_aspect);
WHERE
```

```
WR1: SELF\shape_aspect.description IN ['bonded conductive base blind via',
  'buried via', 'component termination passage',
  'interfacial connection', 'non conductive base blind via',
  'plated conductive base blind via', 'plated cutout',
  'partially plated cutout',
  'plated cutout edge segment',
  'partially plated interconnect module edge',
  'plated interconnect module edge segment',
  'plated interconnect module edge', 'unsupported passage', 'cutout',
  'physical connectivity interrupting cutout',
  'dielectric material passage', 'cutout edge segment',
  'interconnect module edge segment', 'interconnect module edge'];
```

```
WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'DESCRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (it)) AND
  (it\representation_item.name = 'feature of size') AND
  (it\descriptive_representation_item.description IN ['true', 'false']))) = 1)) = 1))
```

```
WR3: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'instantiated template') |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PART_TEMPLATE_DEFINITION' IN
  TYPEOF (it.relateing_shape_aspect)) AND
  (it.relateing_shape_aspect\shape_aspect.description IN
  ['inter stratum feature template',
  'via template',
  'component termination passage template',
  'unsupported passage template']))) = 1;
```

```

WR4: (NOT (SELF\shape_aspect.description = 'cutout edge segment')) OR
(SIZEOF (QUERY (cc <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'composed cutout') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE' IN
TYPEOF (cc.relatng_shape_aspect)) AND
((cc.relatng_shape_aspect\shape_aspect.description =
'cutout') OR
(cc.relatng_shape_aspect\shape_aspect.description =
'physical connectivity interrupting cutout') OR
(cc.relatng_shape_aspect\shape_aspect.description =
'partially plated cutout')))) = 1);

WR5: (NOT (SELF\shape_aspect.description = 'interconnect module edge segment')) OR
(SIZEOF (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'composed edge') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE' IN
TYPEOF (ce.relatng_shape_aspect)) AND
((ce.relatng_shape_aspect\shape_aspect.description =
'interconnect module edge') OR
(ce.relatng_shape_aspect\shape_aspect.description =
'interconnect module edge')
))) = 1);

WR6: (NOT (SELF\shape_aspect.description = 'dielectric material passage')) OR
(SIZEOF (QUERY (pp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'precedent passage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE' IN
TYPEOF (pp.relatng_shape_aspect)) AND
((pp.relatng_shape_aspect\shape_aspect.description =
'cutout') OR
(pp.relatng_shape_aspect\shape_aspect.description =
'physical connectivity interrupting cutout') OR
(pp.relatng_shape_aspect\shape_aspect.description =
'partially plated cutout')))) = 1);

```

```

WR7: (NOT (SELF\shape_aspect.description = 'dielectric material passage')) OR
((SIZEOF (QUERY (rp <* QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY' IN
TYPEOF (pd)) |
rp\property_definition.name = 'feature material')) <= 1) OR
(SIZEOF (USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1));

WR8: (NOT (SELF\shape_aspect.description =
'physical connectivity interrupting cutout')) OR
(SIZEOF (QUERY (ice <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'interrupted connectivity elements') |
ice.relatng_shape_aspect\shape_aspect.name =
'conductive interconnect element')) >= 1);

WR9: (NOT (SELF\shape_aspect.description = 'unsupported passage')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'unsupported passage template')) = 1);

WR10: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION' IN
TYPEOF (SELF.of_shape.definition)) AND
(SIZEOF (QUERY (prpc <*
USEDIN (SELF.of_shape.definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
prpc\product_category.name = 'template model')) >= 1);

WR11: (NOT (SELF\shape_aspect.description = 'cutout edge segment')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)))) = 1);

```



```

WR12: (NOT (SELF\shape_aspect.description =
'plated cutout edge segment')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE' IN
TYPEOF (it.relate_shape_aspect)))) = 1);

WR13: (NOT (SELF\shape_aspect.description =
'interconnect module edge segment')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE' IN
TYPEOF (it.relate_shape_aspect)))) = 1);

WR14: (NOT (SELF\shape_aspect.description =
'plated interconnect module edge segment')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE' IN
TYPEOF (it.relate_shape_aspect)))) = 1);
END_ENTITY; -- inter_stratum_feature
( *

```

Formal propositions:

WR1: The `inter_stratum_feature` shall have a description of 'bonded conductive base blind via', 'buried via', 'component termination passage', 'interfacial connection', 'non conductive base blind via', 'plated conductive base blind via', 'partially plated cutout', 'plated cutout', 'plated cutout edge segment', 'plated interconnect module edge segment', 'partially plated interconnect module edge', 'plated interconnect module edge', 'unsupported passage', 'cutout', 'physical connectivity interrupting cutout', 'dielectric material passage', 'cutout edge segment', 'interconnect module edge segment', or 'interconnect module edge'.

WR2: The `inter_stratum_feature` shall have exactly one `descriptive_representation_item` with a name of 'feature of size' and a description of 'true' or 'false'.

WR3: The `inter_stratum_feature` shall be related, as an instantiated template, to exactly one `part_template_definition` that has a description of 'inter stratum feature template', 'via template', 'component termination passage template', or 'unsupported passage template'.

WR4: If the `inter_stratum_feature` has a description of 'cutout edge segment', the `inter_stratum_feature` shall be related to exactly one `inter_stratum_feature` with a description of either 'cutout' or 'physical connectivity interrupting cutout' or 'partially plated cutout' as the 'composed cutout'.

WR5: If the `inter_stratum_feature` has a description of 'interconnect module edge segment', the `inter_stratum_feature` shall be related to exactly one `inter_stratum_feature` with a description of 'interconnect module edge' or of 'partially plated interconnect module edge' as a 'composed edge'.

WR6: If the `inter_stratum_feature` has a description of 'dielectric material passage', the `inter_stratum_feature` shall be related to exactly one `inter_stratum_feature` with a description of either 'cutout' or 'physical connectivity interrupting cutout' or 'partially plated cutout' as the 'precedent passage'.

WR7: If the `inter_stratum_feature` has a description of 'dielectric material passage', it shall have at most one `requirements_property` with a name of 'feature material' or be the definition for at most one `material_designation`.

WR8: If the `inter_stratum_feature` has a description of 'physical connectivity interrupting cutout', the `inter_stratum_feature` shall be related to at least one `shape_aspect` with a name of 'conductive interconnect element' as 'interrupted connectivity elements'.

WR9: If the `inter_stratum_feature` has a description of 'unsupported passage', the `inter_stratum_feature` shall be related to exactly one `part_template_definition` with a description of 'unsupported passage template' as an 'instantiated template'.

WR10: The `inter_stratum_feature` shall be an aspect of the shape of a `component_definition` that is the definition of a product that is categorized as an 'interconnect module'.

WR11: If the `inter_stratum_feature` has a description of 'cutout edge segment', the `inter_stratum_feature` shall be related, as an 'instantiated template', to exactly one `inter_stratum_feature_edge_segment_template`.

WR12: If the `inter_stratum_feature` has a description of 'plated cutout edge segment', the `inter_stratum_feature` shall be related, as an 'instantiated template', to exactly one `inter_stratum_feature_edge_segment_template`.

WR13: If the `inter_stratum_feature` has a description of 'interconnect module edge segment', the `inter_stratum_feature` shall be related, as an 'instantiated template', to exactly one `inter_stratum_feature_edge_segment_template`.

WR14: If the `inter_stratum_feature` has a description of 'plated interconnect module edge segment', the `inter_stratum_feature` shall be related, as an 'instantiated template', to exactly one `inter_stratum_feature_edge_segment_template`.

5.2.3.1.164 `inter_stratum_feature_edge_segment_template`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `inter_stratum_feature_edge_segment_template` is an `inter_stratum_feature` and `shape_aspect_relationship` that implements the ARM concept of `Interconnect_module_edge_segment` (see 4.2.450).

EXPRESS specification:

```

*)
ENTITY inter_stratum_feature_edge_segment_template
  SUBTYPE OF (part_template_definition, shape_aspect_relationship);
WHERE

  WR1: (SIZEOF (QUERY (ce <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTER_STRATUM_FEATURE_EDGE_TEMPLATE' IN
    TYPEOF (ce.relate_shape_aspect))
    )) = 1);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.relate_shape_aspect);

  WR4: SELF\shape_aspect_relationship.relate_shape_aspect :<:
    SELF\shape_aspect_relationship.related_shape_aspect;

  WR5: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\part_template_definition ||
    SELF\shape_aspect_relationship ||
    SELF\inter_stratum_feature_edge_segment_template)) = 0;
END_ENTITY; -- inter_stratum_feature_edge_segment_template
( *

```

Formal propositions:

WR1: The `inter_stratum_feature_edge_segment_template` shall be related to exactly one `inter_stratum_feature_edge_template`.

WR2: The `inter_stratum_feature_edge_segment_template` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` relating attribute.

WR3: The `inter_stratum_feature_edge_segment_template` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` related attribute.

WR4: The two `edge_segment_vertex` instances referenced shall not be the same instance of `edge_segment_vertex`.

WR5: The `inter_stratum_feature_edge_segment_template` shall not participate in complex instances other than those of its immediate supertypes.

5.2.3.1.165 inter_stratum_feature_edge_template

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `inter_stratum_feature_edge_template` is a `part_template_definition` that implements the ARM concept of `inter_stratum_feature_edge_template` (see 4.2.434).

EXPRESS specification:

```

*)
ENTITY inter_stratum_feature_edge_template
  SUBTYPE OF (part_template_definition);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\part_template_definition ||
    SELF\inter_stratum_feature_edge_template)) = 0;

  WR2: SELF\shape_aspect.description =
    'inter stratum feature edge template';
END_ENTITY; -- inter_stratum_feature_edge_template
( *

```

Formal propositions:

WR1: The `inter_stratum_feature_edge_template` shall not participate in a complex instance of other than its direct supertypes.

WR2: The `inter_stratum_feature_edge_template` shall have a description of 'inter stratum feature edge template'.

5.2.3.1.166 interconnect_definition

An `interconnect_definition` is a `physical_unit` that implements the ARM concepts of `Pcb` (see 4.2.642) and `Pcb_usage_view` (see 4.2.643).

EXPRESS specification:

```

*)
ENTITY interconnect_definition
  SUBTYPE OF (physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

```

```

WR2: NOT EXISTS(SELF\product_definition.name) OR
      (SELF\product_definition.name = 'interconnect module');

WR3: (NOT (SELF.frame_of_reference.name = 'physical design')) OR
      (SIZEOF (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') |
        pdr\product_definition_relationship.name = 'design usage') |
        (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_INTERCONNECT_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_INTERCONNECT_DEFINITION'] *
        TYPEOF (du.relateing_product_definition)) = 1) AND
        (du.relateing_product_definition.frame_of_reference.name =
        'physical design usage') AND
        (du.relateing_product_definition\product_definition.name =
        'interconnect module')
      )) = 1);
END_ENTITY; -- interconnect_definition
(*

```

Formal propositions:

WR1: The `interconnect_definition` shall have a name that is 'interconnect module'.

WR2: If the `interconnect_definition` is a physical design, it shall relate to exactly one `interconnect_definition`, `externally_defined_interconnect_definition`, or `library_defined_interconnect_definition` that is a physical design usage with a name of 'interconnect module' as the design usage.

5.2.3.1.167 `interconnect_module_component_surface_feature`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_component_surface_feature` is a `shape_aspect`. An `interconnect_module_component_surface_feature` implements the ARM concept of: `Interconnect_module_component_surface_feature` (see 4.2.444)

EXPRESS specification:

```

*)
ENTITY interconnect_module_component_surface_feature
  SUBTYPE OF (shape_aspect);
WHERE

```

```

WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_DEFINITION' IN
      TYPEOF (SELF.of_shape.definition);

WR2: (SELF\shape_aspect.description IN [
      'interconnect module component surface feature']);

WR3: SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'instantiated feature') |
      i_f.relate_shape_aspect\shape_aspect.description IN
      ['interconnect module secondary surface',
      'interconnect module primary surface',
      'interconnect module cavity surface',
      'interconnect module cutout surface',
      'interconnect module edge surface',
      'interconnect module edge segment surface'])) = 1;
END_ENTITY; -- interconnect_module_component_surface_feature
(*

```

Formal propositions:

WR1: The `interconnect_module_component_surface_feature` shall be an aspect of the shape of a component_definition.

WR2: The `interconnect_module_component_surface_feature` shall have a description of 'interconnect module component surface feature'.

WR3: The `interconnect_module_component_surface_feature` shall be related to exactly one shape_aspect with a description of 'interconnect module secondary surface', 'interconnect module primary surface', 'interconnect module edge surface', 'interconnect module cavity surface', 'interconnect module cavity surface', or 'interconnect module edge segment surface' as the 'instantiated feature'.

5.2.3.1.168 interconnect_module_design_object_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_design_object_category` is a `characterized_object` that specifies a design object category for the implementation of the ARM concept of `Interconnect_module_constraint_region` (see 4.2.446).

EXPRESS specification:

```

*)
ENTITY interconnect_module_design_object_category
  SUBTYPE OF (characterized_object);
WHERE

  WR1: SELF\characterized_object.description IN
    ['cutout category',
     'fill area category',
     'inter stratum feature category',
     'stratum feature category',
     'via category'];
END_ENTITY; -- interconnect_module_design_object_category
(*)

```

Formal propositions:

WR1: The `interconnect_module_design_object_category` shall have a description of 'cutout category', 'fill area category', 'inter stratum feature category', 'stratum feature category', or 'via category'.

5.2.3.1.169 interconnect_module_cutout_segment_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `interconnect_module_cutout_segment_surface` is a `shape_aspect` and `shape_aspect_relationship` that implements the ARM concept of `Interconnect_module_cutout_segment_surface` (see 4.2.447).

EXPRESS specification:

```

(*)
ENTITY interconnect_module_cutout_segment_surface
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.relatng_shape_aspect);

```

```

WR3: SELF\shape_aspect_relationship.relate_shape_aspect :<>:
      SELF\shape_aspect_relationship.related_shape_aspect;

WR4: sizeof (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'composed surface') |
      ('interconnect module cutout surface' =
      ce.relate_shape_aspect\shape_aspect.description) )) = 1;

WR5: sizeof (typeof (SELF) - typeof (SELF\shape_aspect ||
      SELF\shape_aspect_relationship ||
      SELF\interconnect_module_cutout_segment_surface)) = 0;
END_ENTITY; -- interconnect_module_cutout_segment_surface
(*)

```

Formal propositions:

WR1: The `interconnect_module_cutout_segment_surface` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` related attribute.

WR2: The `interconnect_module_cutout_segment_surface` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` relating attribute.

WR3: The `edge_segment_vertex` instances referenced shall not be the same instance of `edge_segment_vertex`.

WR4: The `interconnect_module_cutout_segment_surface` shall be related to exactly one `shape_aspect` with a description of 'interconnect module cutout surface' as the 'composed surface'.

WR5: The `interconnect_module_cutout_segment_surface` shall not be in a complex instance other than with its immediate supertypes.

5.2.3.1.170 interconnect_module_edge_segment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `interconnect_module_edge_segment` is an `inter_stratum_feature` and `shape_aspect_relationship` that implements the ARM concept of `Interconnect_module_edge_segment` (see 4.2.450).

EXPRESS specification:

```

*)
ENTITY interconnect_module_edge_segment
  SUBTYPE OF (inter_stratum_feature, shape_aspect_relationship);
WHERE

```



```

WR1: SELF\shape_aspect.description IN [
    'plated interconnect module edge segment',
    'interconnect module edge segment'];

WR2: SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'join implementation') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT' IN
    TYPEOF (ji.relatng_shape_aspect)) AND
    (ji.relatng_shape_aspect\shape_aspect.name =
    'inter stratum join')))) <= 1;

WR3: (NOT (SELF\shape_aspect.description =
    'plated interconnect module edge segment')) OR
    (SIZEOF (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'composed edge') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLATED_INTER_STRATUM_FEATURE' IN
    TYPEOF (ce.relatng_shape_aspect)) AND
    (ce.relatng_shape_aspect\shape_aspect.description =
    'plated interconnect module edge')))) = 1);

WR4: (NOT (SELF\shape_aspect.description = 'interconnect module edge segment')) OR
    (SIZEOF (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'composed edge') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTER_STRATUM_FEATURE' IN
    TYPEOF (ce.relatng_shape_aspect)) AND
    (ce.relatng_shape_aspect\shape_aspect.description =
    'interconnect module edge')))) = 1);

WR5: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

WR6: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX' IN
    TYPEOF (SELF\shape_aspect_relationship.relatng_shape_aspect);

WR7: SELF\shape_aspect_relationship.relatng_shape_aspect :<>:
    SELF\shape_aspect_relationship.related_shape_aspect;
END_ENTITY; -- interconnect_module_edge_segment
(

```

Formal propositions:

WR1: The `interconnect_module_edge_segment` shall have a description of either 'plated interconnect module edge segment', or 'interconnect module edge segment'.

WR2: The `interconnect_module_edge_segment` shall be related to at most one `join_shape_aspect` with a name of 'inter stratum join' as a 'join implementation'.

WR3: The `interconnect_module_edge_segment` shall be related to exactly one `plated_inter_stratum_feature` as the 'composed edge' if the description of the `interconnect_module_edge_segment` is 'plated interconnect module edge segment'.

WR4: The `interconnect_module_edge_segment` shall be related to exactly one `inter_stratum_feature` as the 'composed edge'.

WR5: The `interconnect_module_edge_segment` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` relating attribute.

WR6: The `interconnect_module_edge_segment` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` related attribute.

WR7: The `edge_segment_vertex` instances referenced shall not be the same instance of `edge_segment_vertex`.

5.2.3.1.171 `interconnect_module_edge_segment_surface`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `interconnect_module_edge_segment_surface` is a `shape_aspect` and `shape_aspect_relationship` that implements the ARM concept of `Interconnect_module_edge_segment_surface` (see 4.2.451).

EXPRESS specification:

*)

```
ENTITY interconnect_module_edge_segment_surface
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE
```

```
WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX' IN
      TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);
```

```
WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX' IN
      TYPEOF (SELF\shape_aspect_relationship.relate_shape_aspect);
```

```
WR3: SELF\shape_aspect_relationship.relate_shape_aspect :<>:
      SELF\shape_aspect_relationship.related_shape_aspect;
```

```

WR4: sizeof (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'composed surface') |
      ('interconnect module edge surface' =
      ce.relate_shape_aspect\shape_aspect.description) )) = 1;

WR5: sizeof (typeof (SELF) - typeof (SELF\shape_aspect ||
      SELF\shape_aspect_relationship ||
      SELF\interconnect_module_edge_segment_surface)) = 0;
END_ENTITY; -- interconnect_module_edge_segment_surface
(*)

```

Formal propositions:

WR1: The `interconnect_module_edge_segment_surface` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` related attribute.

WR2: The `interconnect_module_edge_segment_surface` shall be related to exactly one `edge_segment_vertex` through a `shape_aspect_relationship` relating attribute.

WR3: The `edge_segment_vertex` instances referenced shall not be the same instance of `edge_segment_vertex`.

WR4: The `interconnect_module_edge_segment_surface` shall be related to exactly one `shape_aspect` with a description of 'interconnect module edge surface' as the 'composed surface'.

WR5: The `interconnect_module_edge_segment_surface` shall not be in a complex instance other than with its immediate supertypes.

5.2.3.1.172 `interconnect_module_interface_terminal`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_interface_terminal` is an `interconnect_module_terminal` that implement the ARM concept of `Interconnect_module_interface_terminal` (see 4.2.453)

EXPRESS specification:

```

*)
ENTITY interconnect_module_interface_terminal
  SUBTYPE OF (interconnect_module_terminal);
END_ENTITY; -- interconnect_module_interface_terminal
(*)

```

5.2.3.1.173 `interconnect_module_join_terminal`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_join_terminal` is an `interconnect_module_terminal` that implement the ARM concept of `Interconnect_module_join_terminal` (see 4.2.454)

EXPRESS specification:

```
*)
ENTITY interconnect_module_join_terminal
  SUBTYPE OF (interconnect_module_terminal);
END_ENTITY; -- interconnect_module_join_terminal
(*
```

5.2.3.1.174 `interconnect_module_macro_component_join_terminal`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_macro_component_join_terminal` is a `shape_aspect` that implements the ARM concept of `internal.ref Linkend="Interconnect_module_macro_component_join_terminal">`

EXPRESS specification:

```
*)
ENTITY interconnect_module_macro_component_join_terminal
  SUBTYPE OF (component_terminal);
WHERE

  WR1: NOT (SELF\shape_aspect.description IN
    ['assembly module component terminal',
     'bare die component terminal',
     'component termination passage join terminal',
     'conductive interconnect element terminal',
     'interconnect component join terminal',
     'land join terminal',
     'minimally defined component terminal',
     'non functional land join terminal',
     'packaged component join terminal',
     'printed component join terminal', 'package terminal occurrence',
     'via terminal']);
END_ENTITY; -- interconnect_module_macro_component_join_terminal
(*
```

Formal propositions:

WR1: The `interconnect_module_macro_component_join_terminal` shall not have a description of 'bare die component terminal', 'component termination passage join terminal', 'conductive interconnect element terminal', 'interconnect component join terminal', 'assembly module component terminal', 'land join terminal', 'minimally defined component terminal', 'non functional land join terminal', 'packaged component join terminal', 'printed component join terminal', 'package terminal occurrence', or 'via terminal'.

5.2.3.1.175 interconnect_module_stratum_based_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `interconnect_module_stratum_based_terminal` is an `interconnect_module_terminal` that implements the ARM concept of `Interconnect_module_stratum_based_terminal` (see 4.2.461).

EXPRESS specification:

```
*)
ENTITY interconnect_module_stratum_based_terminal
  SUBTYPE OF (interconnect_module_terminal);
END_ENTITY; -- interconnect_module_stratum_based_terminal
(*
```

5.2.3.1.176 interconnect_module_terminal

An `interconnect_module_terminal` is a `shape_aspect` that implements the ARM concepts of `Interconnect_module_terminal` (see 4.2.463) and `Part_terminal` (see 4.2.630).

EXPRESS specification:

```
*)
ENTITY interconnect_module_terminal
  SUPERTYPE OF (ONEOF(interconnect_module_interface_terminal,
                      interconnect_module_join_terminal))
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: EXISTS(SELF.of_shape.definition\product_definition.name);
```

```

WR2: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT' IN
      TYPEOF (SELF.of_shape.definition)) AND
      (SELF.of_shape.definition\product_definition.
      frame_of_reference.name =
      'physical design usage') AND
      (SELF.of_shape.definition\product_definition.name =
      'interconnect module');

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF (QUERY (pdr <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION' IN
      TYPEOF (pdr.used_representation))) >= 1)) >= 1;

WR4: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'member connected terminal') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_CONNECTED_TERMINALS_DEFINITION'
      IN TYPEOF (mct.relatng_shape_aspect))) <= 1;

WR5: SIZEOF (QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar.related_shape_aspect\shape_aspect.description =
      'connection zone')) <= 1;
END_ENTITY; -- interconnect_module_terminal
( *

```

Formal propositions:

WR1: The `physical_unit` name shall exist.

WR2: The `interconnect_module_terminal` shall be an aspect of the shape of a `physical_unit` that is an 'interconnect module' and that is a 'physical design usage'.

WR3: The `interconnect_module_terminal` shall have at least one `property_definition` that is represented by a `shape_representation`.

WR4: The `interconnect_module_terminal` shall be related to at most one `part_connected_terminals_definition` as a 'member connected terminal'.

WR5: The `interconnect_module_terminal` shall be related to no more than one `shape_aspect` with a description of 'connection zone'.

5.2.3.1.177 interface_access_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An interface_access_component_shape_aspect is a component_shape_aspect that implements the ARM concept of Interface_access_laminate_component (see ??).

EXPRESS specification:

```
*)
ENTITY interface_access_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
END_ENTITY; -- interface_access_component_shape_aspect
(*
```

5.2.3.1.178 interface_access_material_removal_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An interface_access_material_removal_component_shape_aspect is a material_removal_component_shape_aspect that implements the ARM concept of Interface_access_material_removal_laminate_component (see 4.2.466).

EXPRESS specification:

```
*)
ENTITY interface_access_material_removal_component_shape_aspect
  SUBTYPE OF (material_removal_component_shape_aspect);
END_ENTITY; -- interface_access_material_removal_component_shape_aspect
(*
```

5.2.3.1.179 interface_component_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An interface_component_definition is a component_definition that implements the ARM concept of interface_component (see 4.2.468) .

EXPRESS specification:

```

*)
ENTITY interface_component_definition
  SUBTYPE OF (component_definition);
WHERE

  WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_INTERFACE_TERMINAL' IN
    TYPEOF (sa)) AND
    (sa\shape_aspect.description IN ['interconnect component interface terminal',
    'packaged connector component interface terminal']))) >= 1)) >= 1);
END_ENTITY; -- interface_component_definition
(

```

Formal propositions:

WR1: The interface_component_definition shall have at least one shape_aspect that is a component_interface_terminal with a description of 'interconnect component interface terminal' or 'packaged connector component interface terminal'.

5.2.3.1.180 interface_mounted_join

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An interface_mounted_join is a shape_aspect and a shape_aspect_relationship that relates a component_terminal to a component_interface_terminal. An interface_mounted_join implements the ARM concept of an Interface_mounted_join (see 4.2.470).

EXPRESS specification:

```

*)
ENTITY interface_mounted_join
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
WHERE

```



```

WR1: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_INTERFACE_TERMINAL' IN
      TYPEOF (SELF.related_shape_aspect)) AND
      (SELF.related_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal');

WR2: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL' IN
      TYPEOF (SELF.relate_shape_aspect)) AND
      ((SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal') OR
      (SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal') OR
      (SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal') OR
      (SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal') OR
      (SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal') OR
      (SELF.relate_shape_aspect\shape_aspect.description =
      'packaged connector component interface terminal')));
END_ENTITY; -- interface_mounted_join
(*)

```

Formal propositions:

WR1: The interface_mounted_join shall have a 'packaged connector component interface terminal' as the related_shape_aspect.

WR2: The interface_mounted_join shall have a component_terminal with the listed description as the relate_shape_aspect.

5.2.3.1.181 interfaced_group_component_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An interfaced_group_component_definition is a assembly_group_component_definition that implements the ARM concept of Interfaced_group_component (see 4.2.473) .

EXPRESS specification:

```

*)
ENTITY interfaced_group_component_definition
  SUBTYPE OF (assembly_group_component_definition);
WHERE

```

```

WR1: (SIZEOF (QUERY (gc <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'group component') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERFACE_COMPONENT_DEFINITION' IN
    TYPEOF (gc.related_product_definition))))
    >= 1);
END_ENTITY; -- interfaced_group_component_definition
(*)

```

Formal propositions:

WR1: The `interfaced_group_component_definition` shall relate to at least one `interface_component_definition` to specify a 'group component'.

5.2.3.1.182 join_shape_aspect

A `join_shape_aspect` is a `shape_aspect` that implements the ARM concepts of:

- `Inter_stratum_join_relationship` (see 4.2.437);
- `Intra_stratum_join_relationship` (see 4.2.477);
- `Join_relationship` (see 4.2.479);
- `Unrouted_join_relationship` (see 4.2.938).

EXPRESS specification:

```

*)
ENTITY join_shape_aspect
  SUBTYPE OF (shape_aspect);
WHERE

```

```

WR1: NOT(SELF\shape_aspect.name IN ['constrained intra layer join',
    'inter stratum join', 'intra stratum join']) OR
    (SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'connected point') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYER_CONNECTION_POINT' IN
    TYPEOF (cp.related_shape_aspect)))) >= 2);

```

```

WR2: SIZEOF (QUERY (nt <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'network topology') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_NETWORK' IN
TYPEOF (nt.relate_shape_aspect))) = 1;

WR3: (NOT (SELF\shape_aspect.name = 'intra stratum join')) OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE' IN
TYPEOF (ji.related_shape_aspect)) AND
(ji.related_shape_aspect\shape_aspect.description = 'conductor')))) <= 1);

WR4: (NOT (SELF\shape_aspect.name = 'intra stratum join')) OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE' IN
TYPEOF (ji.related_shape_aspect)) AND
(ji.related_shape_aspect\shape_aspect.description = 'connected filled area'))))
<= 1);

WR5: (NOT (SELF\shape_aspect.name = 'inter stratum join')) OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_CONCEPT_RELATIONSHIP' IN
TYPEOF (ji.related_shape_aspect)) AND
(ji.related_shape_aspect\shape_aspect.description =
'physical network supporting stratum feature conductive join')) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_INTER_STRATUM_FEATURE' IN
TYPEOF (ji.related_shape_aspect)))) <= 1);

```

```

WR6: NOT(SELF\shape_aspect.name = 'unrouted join') OR
      (SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'unrouted terminals') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL' IN
        TYPEOF (cp.related_shape_aspect)))) >= 2);
END_ENTITY; -- join_shape_aspect
(*)

```

Formal propositions:

WR1: The join_shape_aspect with a name of 'constrained intra layer join', 'inter stratum join', or 'intra stratum join' shall be used to relate at least two layer_connection_points to specify a 'connected point'.

WR2: The join_shape_aspect shall be related to exactly one physical_network as the 'network topology'.

WR3: If the join_shape_aspect has a name of 'intra stratum join', it shall relate to at most one stratum_feature with a description of 'conductor' as a 'join implementation'.

WR4: If the join_shape_aspect has a name of 'intra stratum join', it shall relate to at most one stratum_feature with a description of 'connected filled area' as a 'join implementation'.

WR5: If the join_shape_aspect has a name of 'inter stratum join', it shall relate to at most one stratum_concept_relationship with a description of 'physical network supporting stratum feature conductive join' or plated_inter_stratum_feature as a 'join implementation'.

WR6: The join_shape_aspect with a name of 'unrouted join', shall be used to relate at least two component_terminal to specify 'unrouted terminals'.

5.2.3.1.183 keepout_design_object_category

A keepout_design_object_category is a characterized_object that specifies a design object category for the implementation of the ARM concepts of Part_template_3d_keepout_shape (see 4.2.622) , Part_template_planar_keepout_shape (see 4.2.629) , Physical_unit_3d_keepout_shape (see 4.2.665) , and Physical_unit_planar_keepout_shape (see 4.2.669) .

EXPRESS specification:

```

*)
ENTITY keepout_design_object_category
  SUBTYPE OF (characterized_object);
WHERE

```

```

WR1: SELF\characterized_object.description IN
    ['assembly module assembly component category',
     'component feature category',
     'assembly ee material category',
     'interconnect ee material category',
     'interconnect module assembly component category',
     'via category',
     'inter stratum feature category',
     'cutout category',
     'fill area category',
     'lamine component category',
     'stratum feature category'];
END_ENTITY; -- keepout_design_object_category
(*)

```

Formal propositions:

WR1: The keepout_design_object_category shall have a description of 'assembly module assembly component category', 'component feature category', 'assembly ee material category', 'interconnect ee material category', 'interconnect module assembly component category', 'via category', 'inter stratum feature category', 'cutout category', 'fill area category', 'lamine component category', or 'stratum feature category'.

5.2.3.1.184 laminate_component_interface_terminal

A laminate_component_interface_terminal is a shape_aspect. A laminate_component_interface_terminal implements the ARM concepts of:

- Component_termination_passage_interface_terminal (see 4.2.148);
- Land_interface_terminal (see 4.2.491);
- Non_functional_land_interface_terminal (see 4.2.563);
- Printed_connector_component_interface_terminal (see 4.2.725).

EXPRESS specification:

```

*)
ENTITY laminate_component_interface_terminal
    SUBTYPE OF (shape_aspect);
WHERE

```

```

WR1: (NOT (SELF\shape_aspect.description IN
    ['component termination passage interface terminal',
     'land interface terminal',
     'printed connector component interface terminal',
     'non functional land interface terminal'])) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'COMPONENT_DEFINITION' IN
     TYPEOF (SELF.of_shape.definition));

WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature')) = 1;
(*

WR3: (NOT (SELF\shape_aspect.description =
    'component termination passage interface terminal')) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLATED_PASSAGE' IN
    TYPEOF (associated_component(SELF))) AND
    (associated_component(SELF).description =
    'component termination passage'));
*)

WR4: (NOT (SELF\shape_aspect.description =
    'component termination passage interface terminal')) OR
    (SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature') |
    i_f.relate_shape_aspect\shape_aspect.description =
    'component termination passage template interface terminal')) = 1);

WR5: (NOT (SELF\shape_aspect.description IN
    ['land interface terminal',
     'non functional land interface terminal'])) OR
    (SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated feature') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND_TEMPLATE_TERMINAL' IN
    TYPEOF (i_f.relate_shape_aspect)) AND
    (TRUE)))) = 1);
(*)

```

```

WR6: NOT(SELF\shape_aspect.description = 'land interface terminal') OR
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'LAND' IN
        TYPEOF (associated_component(SELF))) AND
(associated_component(SELF).description IN
  ['via and contact size dependent land',
   'contact size dependent land',
   'component termination passage and contact size dependent land']));

WR7: NOT(SELF\shape_aspect.description =
        'non functional land interface terminal') OR
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'LAND' IN
        TYPEOF (associated_component(SELF))) AND
(associated_component(SELF).description IN
  ['via and contact size dependent non functional land',
   'contact size dependent non functional land',
   'component termination passage and contact size dependent non functional land']));
*)

WR8: (NOT (SELF\shape_aspect.description =
          'printed connector component interface terminal')) OR
      (SELF.product_definitional);

WR9: (NOT (SELF\shape_aspect.description =
          'printed connector component interface terminal')) OR
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
       'PRINTED_COMPONENT' IN
        TYPEOF (SELF.of_shape.definition)) AND
      (SELF.of_shape.definition\product_definition.description =
       'printed connector component'));

WR10: (NOT (SELF\shape_aspect.description =
          'printed connector component interface terminal')) OR
      (SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          sar\shape_aspect_relationship.name = 'instantiated feature') |
          ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRINTED_PART_TEMPLATE_TERMINAL' IN
           TYPEOF (i_f.relatng_shape_aspect)) AND
          (i_f.relatng_shape_aspect\shape_aspect.description =
           'interface terminal')) = 1));

```

```

WR11: (NOT (SELF\shape_aspect.description =
'printed connector component interface terminal')) OR
(SIZEOF (QUERY (i <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
pdr\shape_aspect_relationship.name = 'implementation') |
(SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE'] *
TYPEOF (i.relateing_shape_aspect)) = 1)
)) = 1);

WR12: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION' IN
TYPEOF (SELF.of_shape.definition)) AND
(SELF.of_shape.definition\
product_definition.frame_of_reference\
application_context_element.name = 'layout occurrence');
END_ENTITY; -- laminate_component_interface_terminal
(*

```

Formal propositions:

WR1: The laminate_component_interface_terminal shall be an aspect of the shape of a component_definition.

WR2: The laminate_component_interface_terminal shall be related to exactly one shape_aspect as an 'instantiated feature'.

WR3: If the laminate_component_interface_terminal has a description of 'component termination passage interface terminal', the laminate_component_interface_terminal shall be related to exactly one plated_passage with a description of 'component termination passage' as the 'associated component'.

WR4: If the laminate_component_interface_terminal has a description of 'component termination passage interface terminal', the laminate_component_interface_terminal shall be related to exactly one shape_aspect with a description of 'component termination passage template interface terminal' as the 'instantiated feature'.

WR5: If the laminate_component_interface_terminal has a description of either 'land interface terminal' or 'non functional land interface terminal', the laminate_component_interface_terminal shall be related to exactly one land_template_terminal as the 'instantiated feature'.

WR6: If the laminate_component_interface_terminal has a description of 'land interface terminal', the laminate_component_interface_terminal shall be related to exactly one land with the appropriate description as the 'associated component'.

WR7: If the laminate_component_interface_terminal has a description of 'non functional land interface terminal', the laminate_component_interface_terminal shall be related to exactly one land with the appropriate description as the 'associated component'.

WR8: If the `laminate_component_interface_terminal` has a description of 'printed connector component interface terminal', it shall lie on the boundary of a product.

WR9: If the `laminate_component_interface_terminal` has a description of 'printed connector component interface terminal', it shall be an aspect of the shape of a `printed_component` that has a description of 'printed connector component'.

WR10: If the `laminate_component_interface_terminal` has a description of 'printed connector component interface terminal', the `laminate_component_interface_terminal` shall be related to exactly one `printed_part_template_terminal` with a description of 'interface terminal' as the 'instantiated feature'.

WR11: If the `laminate_component_interface_terminal` has a description of 'printed connector component interface terminal', the `laminate_component_interface_terminal` shall be related to exactly one `stratum_feature` as the 'implementation'.

WR12: The `laminate_component_interface_terminal` shall be a `shape_aspect` for a `component_definition` with a `frame_of_reference` of 'layout occurrence'.

5.2.3.1.185 `laminate_group_component_make_from_relationship`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `laminate_group_component_make_from_relationship` is a `make_from_usage_option` that implements the ARM concept of `Laminate_group_component_make_from_relationship` (see 4.2.487).

EXPRESS specification:

*)

ENTITY `laminate_group_component_make_from_relationship`

SUBTYPE OF (`make_from_usage_option`);

WHERE

WR1: SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_GROUP_COMPONENT_DEFINITION'] *
TYPEOF (SELF.relying_product_definition)) = 1;

WR2: SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_GROUP_COMPONENT_DEFINITION'] *
TYPEOF (SELF.related_product_definition)) = 1;

```

WR3: sizeof( QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (sizeof( QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ((pdr.name = 'stratum mapping') AND
    (pdr.related_property_definition\property_definition.
    definition\product_definition_relationship.name = 'stratum make from')
    AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAKE_FROM_USAGE_OPTION' IN
    TYPEOF(pdr.related_property_definition\property_definition.definition)))
    )) > 0)
    )) > 0;
END_ENTITY; -- laminate_group_component_make_from_relationship
(*

```

Formal propositions:

WR1: The `laminate_group_component_make_from_relationship` shall reference an `assembly_group_component_definition` as its `relating_product_definition`.

WR2: The `laminate_group_component_make_from_relationship` shall reference an `assembly_group_component_definition` as its `related_product_definition`.

WR3: The `laminate_group_component_make_from_relationship` shall reference at least one `make_from_usage_option` with a name of 'stratum make from' as the 'stratum mapping'.

5.2.3.1.186 land

A land is a `component_shape_aspect` that implements the ARM concepts of:

- `Component_termination_passage_and_contact_size_dependent_land` (see 4.2.144);
- `Component_termination_passage_and_contact_size_dependent_non_functional_land` (see 4.2.145);
- `Component_termination_passage_dependent_land` (see 4.2.146);
- `Contact_size_dependent_land` (see 4.2.194);
- `Contact_size_dependent_non_functional_land` (see 4.2.195);
- `Non_functional_land` (see 4.2.562);
- `Land` (see 4.2.490);

- Unsupported_passage_dependent_non_functional_land (see 4.2.940);
- Via_and_contact_size_dependent_land (see 4.2.948);
- Via_and_contact_size_dependent_non_functional_land (see 4.2.949);
- Via_dependent_land (see 4.2.950);
- Via_dependent_non_functional_land (see 4.2.951).

EXPRESS specification:

```

*)
ENTITY land
  SUBTYPE OF (component_shape_aspect);
WHERE

  WR1: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition)) AND
        (SIZEOF (QUERY (prpc <*
        USEDIN (SELF.of_shape.definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
        prpc\product_category.name = 'template model')) >= 1);

  WR2: SIZEOF (QUERY (i_f <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'instantiated template') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_PHYSICAL_TEMPLATE' IN
        TYPEOF (i_f.relate_shape_aspect))) = 1;

  WR3: SIZEOF (QUERY (aud <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'alternate usage definition') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_PHYSICAL_TEMPLATE' IN
        TYPEOF (aud.relate_shape_aspect))) <= 1;

```

```

WR4: SIZEOF (QUERY (sfi <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'stratum feature implementation') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' IN
    TYPEOF (sfi.relatng_shape_aspect)))) = 1;
(*

WR5: (NOT (SELF\shape_aspect.description IN
    ['via and contact size dependent land',
    'component termination passage and contact size dependent land'])) OR
    (SIZEOF (QUERY (at <* associated_terminals(SELF) |
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (at)) AND
        (at\shape_aspect.description =
        'land interface terminal')))) >= 0);

WR6: (NOT (SELF\shape_aspect.description IN
    ['functional land', 'via dependent land',
    'via and contact size dependent land',
    'component termination passage dependent land',
    'contact size dependent land',
    'component termination passage and contact size dependent land'])) OR
    (SIZEOF (QUERY (at <* associated_terminals(SELF) |
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL' IN
        TYPEOF (at)) AND
        (at\shape_aspect.description =
        'land join terminal')))) >= 1);

WR7: (NOT (SELF\shape_aspect.description IN
    ['via and contact size dependent non functional land',
    'component termination passage and contact size '])) OR
    (SIZEOF (QUERY (at <* associated_terminals(SELF) |
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (at)) AND
        (at\shape_aspect.description =
        'non functional land interface terminal')))) = 1);

```

```

WR8: (NOT (SELF\shape_aspect.description IN
['non functional land', 'via dependent non functional land',
'via and contact size dependent non functional land',
'component termination passage dependent non functional land',
'contact size dependent non functional land',
'component termination passage and contact size ' +
'dependent non functional land',
'unsupported passage dependent non functional land'])) OR
(SIZEOF (QUERY (at <* associated_terminals(SELF)|
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL' IN
TYPEOF (at)) AND
(at\shape_aspect.description =
'non functional land join terminal')))) >= 1);

*)

WR9: (NOT (SELF\shape_aspect.description IN
['component termination passage and contact size dependent land',
'component termination passage and contact size dependent ' +
'non functional land',
'component termination passage dependent land',
'component termination passage dependent non functional land'])) OR
(SIZEOF (QUERY (rp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference passage') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE' IN
TYPEOF (rp.relatng_shape_aspect)) AND
(rp.relatng_shape_aspect\shape_aspect.description =
'component termination passage')))) = 1);

WR10: (NOT (SELF\shape_aspect.description IN
['via and contact size dependent land',
'via and contact size dependent non functional land',
'via dependent land', 'via dependent non functional land'])) OR
(SIZEOF (QUERY (rv <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference via') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE' IN
TYPEOF (rv.relatng_shape_aspect)) AND
(rv.relatng_shape_aspect\shape_aspect.description IN
['buried via', 'interfacial connection',
'bonded conductive base blind via', 'non conductive base blind via',
'plated conductive base blind via']))) = 1);

```

```

WR11: (NOT (SELF\shape_aspect.description =
'unsupported passage dependent non functional land')) OR
(SIZEOF (QUERY (rp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference passage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE' IN
TYPEOF (rp.relatng_shape_aspect)) AND
(rp.relatng_shape_aspect\shape_aspect.description =
'unsupported passage')))) = 1);

WR12: (NOT (SELF\shape_aspect.description IN
['component termination passage dependent land',
'component termination passage dependent non functional land'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'default component termination passage based')))) = 1);

WR13: (NOT (SELF\shape_aspect.description IN
['component termination passage and contact size dependent land',
'component termination passage and contact size dependent ' +
'non functional land'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'default attachment size and component termination ' +
'passage based')))) = 1);

```

```

WR14: (NOT (SELF\shape_aspect.description IN
['contact size dependent land',
'contact size dependent non functional land'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'default attachment size based')))) = 1);

WR15: (NOT (SELF\shape_aspect.description IN
['via and contact size dependent land',
'via and contact size dependent non functional land'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'default attachment size and via based')))) = 1);

WR16: (NOT (SELF\shape_aspect.description IN
['via dependent land',
'via dependent non functional land'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE' IN
TYPEOF (it.relatng_shape_aspect)) AND
(it.relatng_shape_aspect\shape_aspect.description =
'default via based')))) = 1);

```

```

WR17: (NOT (SELF\shape_aspect.description =
    'unsupported passage dependent ' +
    'non functional land')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated template') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND_PHYSICAL_TEMPLATE' IN
    TYPEOF (it.relateing_shape_aspect)) AND
    (it.relateing_shape_aspect\shape_aspect.description =
    'default unsupported passage based')))) = 1);
END_ENTITY; -- land
( *

```

Formal propositions:

WR1: The land shall be an aspect of the shape of a component_definition that is the definition of a product that is categorized as an 'template module'.

WR2: The land shall be related to exactly one land_physical_template as the 'instantiated template'.

WR3: The land shall be related to at most one land_physical_template as the 'alternate usage definition'.

WR4: The land shall be related to exactly one stratum_feature as a 'stratum feature implementation'.

WR5: If the land has a description of 'via and contact size dependent land', or 'component termination passage and contact size dependent land', the land may relate to zero or more component_shape_aspect that has a description of 'land interface terminal' to specify the 'associated component'.

NOTE A contact size dependent land may have more than one interface terminal.

WR6: If the land has a description of 'functional land', 'via dependent land', 'via and contact size dependent land', 'component termination passage dependent land', 'contact size dependent land', or 'component termination passage and contact size dependent land', the land shall relate to at least one component_terminal that has a description of 'land join terminal' to specify the 'associated component'.

WR7: If the land has a description of 'via and contact size dependent non functional land', 'component termination passage and contact size dependent non functional land', the land shall relate to exactly one component_shape_aspect that has a description of 'non functional land interface terminal' to specify the 'associated component'.

WR8: If the land has a description of 'non functional land', 'via dependent non functional land', 'via and contact size dependent non functional land', 'component termination passage dependent non functional land', 'contact size dependent non functional land', 'component termination passage and contact size dependent non functional land', or 'unsupported passage dependent non functional land', the land shall relate to at least one component_terminal that has a description of 'non functional land join terminal' to specify the 'associated component'.

WR9: If the land has a description of 'component termination passage and contact size dependent land', 'component termination passage and contact size dependent non functional land', 'component termination passage dependent land', or 'component termination passage dependent non functional land', the land shall be related to exactly one plated_passage with a description of 'component termination passage' as a 'reference passage'.

WR10: If the land has a description of 'via and contact size dependent land', 'via and contact size dependent non functional land', 'via dependent land', or 'via dependent non functional land', the land shall be related to exactly one plated_passage that has a description of 'buried via', 'interfacial connection', 'bonded conductive base blind via', 'non conductive base blind via', or 'plated conductive base blind via' as a 'reference via'.

WR11: If the land has a description of 'unsupported passage dependent non functional land', the land shall be related to exactly one inter_stratum_feature with a description of 'unsupported passage' as a 'reference passage'.

WR12: If the land has a description of either 'component termination passage dependent land' or 'component termination passage dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default component termination passage based' as an 'instantiated template'.

WR13: If the land has a description of either 'component termination passage and contact size dependent land' or 'component termination passage and contact size dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default attachment size and component termination passage based' as an 'instantiated template'.

WR14: If the land has a description of either 'contact size dependent land' or 'contact size dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default attachment size based' as an 'instantiated template'.

WR15: If the land has a description of either 'via and contact size dependent land' or 'via and contact size dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default attachment size and via based' as an 'instantiated template'.

WR16: If the land has a description of either 'via dependent land' or 'via dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default via based' as an 'instantiated template'.

WR17: If the land has a description of 'unsupported passage dependent non functional land', the land shall be related to exactly one land_physical_template with a description of 'default unsupported passage based' as an 'instantiated template'.

5.2.3.1.187 land_physical_template

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A land_physical_template is a part_template_definition that implements the ARM concepts of:

- Default_attachment_size_and_component_termination_passage_based_land_physical_template (see 4.2.245);
- Default_attachment_size_and_via_based_land_physical_template (see 4.2.246);
- Default_attachment_size_based_land_physical_template (see 4.2.247);
- Default_component_termination_passage_based_land_physical_template (see 4.2.248);
- Default_land_physical_template (see 4.2.250);
- Default_passage_based_land_physical_template (see 4.2.251);
- Default_via_based_land_physical_template (see 4.2.255);
- Land_physical_template (see 4.2.493).

EXPRESS specification:

```

*)
ENTITY land_physical_template
  SUBTYPE OF (part_template_definition);
WHERE

  WR1: SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    SIZEOF (QUERY (duc <* USEDIN (ada.assigned_document,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    duc\document_usage_constraint.subject_element =
      'pre defined classification code')) = 1)) <= 1;

  WR2: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated definition') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND_TEMPLATE_TERMINAL' IN
    TYPEOF (am.related_shape_aspect))) >= 1;

```

```

WR3: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN ['default attachment size based',
'default attachment size and component termination passage based',
'default attachment size and via based'])) OR
(SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
((sar\shape_aspect_relationship.name = 'minimum attachment region size') AND
(sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) <= 1));

WR4: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN ['default attachment size based',
'default attachment size and component termination passage based',
'default attachment size and via based'])) OR
(SIZEOF (QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
((sar\shape_aspect_relationship.name = 'maximum attachment region size') AND
(sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) <= 1));

WR5: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default component termination passage based',
'default attachment size and component termination passage based'])) OR
(SIZEOF (QUERY (tu <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'technology usage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN
TYPEOF (tu.related_shape_aspect)) AND
(tu.related_shape_aspect\shape_aspect.description =
'default component termination passage definition')) = 1));

WR6: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default attachment size and component termination passage based',
'default attachment size based',
'default attachment size and via based',
'default component termination passage based',
'default via based',
'default unsupported passage based'])) OR
(SIZEOF (QUERY (ost <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'of stratum technology') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY' IN
TYPEOF (ost.relate_shape_aspect.of_shape.definition)
)) = 1));

```

```

WR7: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based', 'default via based',
'default unsupported passage based'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2)) = 2)) = 1))) = 0));

```

```

WR8: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based', 'default via based',
'default unsupported passage based'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name = 'minimum fabrication allowance'))
= 1)) = 1))) = 0));

```

```

WR9: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based', 'default via based',
'default unsupported passage based'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name = 'minimum annular ring')))) = 1
)) = 1))) = 0));

WR10: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN
['default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based', 'default via based',
'default unsupported passage based'])) OR
(SIZEOF (QUERY (tu <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'technology usage') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN
TYPEOF (tu.related_shape_aspect))) = 1));

WR11: NOT(EXISTS(SELF\shape_aspect.description)) OR
((NOT (SELF\shape_aspect.description IN ['default via based',
'default attachment size and via based'])) OR
(SIZEOF (QUERY (tu <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'technology usage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN
TYPEOF (tu.related_shape_aspect)) AND
(tu.related_shape_aspect\shape_aspect.description = 'default via definition'))
= 1));

```

```

WR12: NOT(EXISTS(SELF\shape_aspect.description)) OR
      ((NOT (SELF\shape_aspect.description = 'default unsupported passage')) OR
      (SIZEOF (QUERY (tu <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'technology usage') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PASSAGE_TECHNOLOGY' IN
      TYPEOF (tu.related_shape_aspect)) AND
      (tu.related_shape_aspect\shape_aspect.description =
      'default unsupported passage definition')))) = 1));
END_ENTITY; -- land_physical_template
(*

```

Formal propositions:

WR1: The land_physical_template shall have a reference of at most one document that is constrained by a 'pre defined classification code'.

WR2: The land_physical_template shall be used to relate at least one land_template_terminal to specify the 'access mechanism'.

WR3: If the land_physical_template has a description of 'default attachment size based', 'default attachment size and component termination passage based', or 'default attachment size and via based', it shall be to relate no more than one shape_aspect with a name of 'connection zone' as the 'minimum attachment region size'.

WR4: If the land_physical_template has a description of 'default attachment size based', 'default attachment size and component termination passage based', or 'default attachment size and via based', it shall be to relate no more than one shape_aspect with a name of 'connection zone' as the 'maximum attachment region size'.

WR5: If the land_physical_template has a description of 'default component termination passage based' or 'default attachment size and component termination passage based', it shall relate to a passage_technology with a description of 'default component termination passage definition' to specify the 'technology usage'.

WR6: If the land_physical_template has a description of 'default attachment size and component termination passage based', 'default attachment size based', 'default attachment size and via based', 'default component termination passage based', 'default via based', or 'default unsupported passage based', it shall relate to a stratum_technology as the 'of stratum technology'.

WR7: If the land_physical_template has a description of 'default attachment size and component termination passage based', 'default attachment size and via based', 'default component termination passage based', 'default via based', or 'default unsupported passage based', it shall have a representation that contains exactly two length_measure_with_units.

WR8: If the `land_physical_template` has a description of 'default attachment size and component termination passage based', 'default attachment size and via based', 'default component termination passage based', 'default via based', or 'default unsupported passage based', it shall have a representation that contains exactly one `length_measure_with_unit` with a name of 'minimum fabrication allowance'.

WR9: If the `land_physical_template` has a description of 'default attachment size and component termination passage based', 'default attachment size and via based', 'default component termination passage based', 'default via based', or 'default unsupported passage based', it shall have a representation that contains exactly one `length_measure_with_unit` with a name of 'minimum annular ring'.

WR10: If the `land_physical_template` has a description of 'default attachment size and component termination passage based', 'default attachment size and via based', 'default component termination passage based', 'default via based', or 'default unsupported passage based', it shall relate a `passage_technology` to specify the 'technology usage'.

WR11: If the `land_physical_template` has a description of 'default via based' or 'default attachment size and via based', it shall relate to a `passage_technology` with a description of 'default via definition' to specify the 'technology usage'.

WR12: If the `land_physical_template` has a description of 'default unsupported passage', it shall relate to a `passage_technology` with a description of 'default unsupported passage definition' to specify the 'technology usage'.

5.2.3.1.188 `land_template_terminal`

A `land_template_terminal` is a `shape_aspect` that implements the ARM concepts of:

- `Land_template_interface_terminal` (see 4.2.494);
- `Land_template_join_terminal` (see 4.2.495);
- `Land_template_terminal` (see 4.2.496).

EXPRESS specification:

```
*)
ENTITY land_template_terminal
  SUBTYPE OF (shape_aspect);
WHERE
```

```
  WR1: SELF\shape_aspect.description IN ['interface terminal', 'join terminal'];
```

```

WR2: SIZEOF (QUERY (tcz <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'terminal connection zone') AND
    (sar\shape_aspect_relationship.description IN ['edge curve', 'edge point',
    'surface area', 'surface point']))) |
    tcz.related_shape_aspect\shape_aspect.description = 'connection zone')) >= 0;
END_ENTITY; -- land_template_terminal
(*

```

Formal propositions:

WR1: The land_template_terminal shall have a description of either 'interface terminal' or 'land terminal'.

WR2: The land_template_terminal shall relate to zero or more shape_aspect with a name of 'connection zone' to specify the 'terminal connection zone' which may be described as a 'edge curve', 'edge point', 'surface_area', or 'surface point'.

5.2.3.1.189 layer

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A layer is a shape_aspect that implements the ARM concept of Layer (see 4.2.498).

EXPRESS specification:

```

*)
ENTITY layer
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1: SELF\shape_aspect.name;
  WHERE

  WR1: SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    SIZEOF (QUERY (duc <* USEDIN (ada.assigned_document,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    duc\document_usage_constraint.subject_element = 'layer definition')) = 1)) <= 1;

```



```

WR2: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP'
      IN TYPEOF (SELF.of_shape.definition)) AND
      (SELF.of_shape.definition\product_definition_relationship.name =
        'inter stratum extent');
END_ENTITY; -- layer
(*)

```

Formal propositions:

UR1: The name shall be unique within a population of layer.

WR1: The layer shall be referenced by at most one document that has a usage constraint which specifies the 'layer definition'.

WR2: The layer shall be an aspect of the shape of a product_definition_relationship with a name of 'inter stratum extent'.

Informal propositions:

IP1: The same geometric representation shall be attached to each of the stratum specified by the layer.

IP2: The name shall be unique.

5.2.3.1.190 layer_connection_point

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A layer_connection_point is a shape_aspect that implements the ARM concepts of:

- Dependently_located_layer_connection_point (see 4.2.259);
- Explicitly_located_layer_connection_point (see 4.2.360);
- Layer_connection_point (see 4.2.499).

EXPRESS specification:

```

*)
ENTITY layer_connection_point
  SUBTYPE OF (shape_aspect);
UNIQUE
  UR1: SELF\shape_aspect.name, SELF\shape_aspect.of_shape;
WHERE

```

WR1: SELF\shape_aspect.of_shape.definition\product_definition.
name = 'design layer';

WR2: SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'connected point') |
((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT' IN
TYPEOF (cp.relating_shape_aspect)) AND
(cp.relating_shape_aspect.name = 'inter stratum join')))) <= 1;

WR3: SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'connected point') |
((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT' IN
TYPEOF (cp.relating_shape_aspect)) AND
(cp.relating_shape_aspect.name = 'intra stratum join')))) = 1;

WR4: NOT(
SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'connected point') |
((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT' IN
TYPEOF (cp.relating_shape_aspect)) AND
(cp.relating_shape_aspect.name = 'inter stratum join')))) = 1)
OR
(SIZEOF (QUERY (cp <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'connected point') |
((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT' IN
TYPEOF (cp.relating_shape_aspect)) AND
(cp.relating_shape_aspect.name = 'intra stratum join')))) = 1);

```

WR5: (NOT (SELF\shape_aspect.description = 'dependently located')) OR
(SIZEOF (QUERY (ado <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'associated design object') |
(SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRUTUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_TERMINAL'] *
TYPEOF (ado.relatng_shape_aspect)) = 1) OR
((SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL'] *
TYPEOF (ado.relatng_shape_aspect)) = 1) AND
(ado.relatng_shape_aspect.description IN ['printed component join terminal',
'embedded component terminal'])))
)) = 1);

WR6: (SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation)) AND
(pdr.used_representation\representation.name = 'connection point location 2d') AND
(SIZEOF (QUERY (it <* pdr.used_representation.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT' IN
TYPEOF (it)) AND (it\geometric_representation_item.dim = 2 )
)) = 1))) <= 1))) = 0);

WR7: (SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation)) AND
(pdr.used_representation\representation.name = 'connection point location 3d') AND
(SIZEOF (QUERY (it <* pdr.used_representation.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT' IN
TYPEOF (it)) AND (it\geometric_representation_item.dim = 3 )
)) = 1))) <= 1))) = 0);

```

```

WR8: (SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation)) AND
((pdr.used_representation\representation.name = 'connection point location 2d')
OR
(pdr.used_representation\representation.name = 'connection point location 3d'))
)) >= 1))) = 0);

WR9: (NOT (SELF\shape_aspect.description = 'dependently located')) OR
(SIZEOF (QUERY (ado <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'reference zone') |
(SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT' ] *
TYPEOF (ado.relatng_shape_aspect)) = 1) OR
((SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT' ] *
TYPEOF (ado.relatng_shape_aspect)) = 1) AND
(ado.relatng_shape_aspect.description IN ['connection zone'])))
)) <= 1);
END_ENTITY; -- layer_connection_point
(*

```

Formal propositions:

UR1: The name shall be unique within the stratum that is the product_definition for this layer_connection_point.

WR1: The layer_connection_point shall be an aspect of the shape of a product_definition with a name of 'design layer'.

WR2: The layer_connection_point shall be related to no more than one join_shape_aspect with a name of 'inter stratum join' as a 'connected point'.

WR3: The layer_connection_point shall be related to one join_shape_aspect with a name of 'intra stratum join' as a 'connected point'.

WR4: If the layer_connection_point is related to a join_shape_aspect with a name of 'inter stratum join' as a 'connected point', the layer_connection_point shall be related to a join_shape_aspect with a name of 'intra stratum join' as a 'connected point'.

WR5: If the `layer_connection_point` has a description of 'dependently located', it shall be related to exactly one `inter_stratum_feature` or `interconnect_module_terminal` or `component_terminal` that has a description of 'embedded component terminal' or 'printed component join terminal' as the 'associated design object'.

WR6: The `layer_connection_point` shall have no more than one `shape_representation` as the 'connection point location 2d' that contains exactly one `cartesian_point` with a dimensionality of 2.

WR7: The `layer_connection_point` shall have no more than one `shape_representation` as the 'connection point location 3d' that contains exactly one `cartesian_point` with a dimensionality of 3.

WR8: The `layer_connection_point` shall have no less than one `shape_representation` as either the 'connection point location 2d' or the 'connection point location 3d'.

WR9: If the `layer_connection_point` has a description of 'dependently located', it shall be related to no more than one `shape_aspect` that has a description of 'connection zone' as the 'reference zone'.

Informal propositions:

IP1: Each `layer_connection_point` shall be within the resolved areal extent of not more than one `stratum_feature`.

IP2: Each `layer_connection_point` shall be specified by not more than one `physical_network`.

5.2.3.1.191 layer_connection_point_link

A `layer_connection_point_link` is a `shape_aspect` and a `shape_aspect_relationship` that implements the ARM concept of `Layer_connection_point_link` (see 4.2.500)

EXPRESS specification:

```
*)
ENTITY layer_connection_point_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAYER_CONNECTION_POINT' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAYER_CONNECTION_POINT' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);
```

```

WR4: sizeof(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\shape_aspect_relationship ||
    SELF\layer_connection_point_link)) = 0;

WR5: SELF\shape_aspect.name = '';

WR6: SELF\shape_aspect_relationship.name = '';

WR7: sizeof(QUERY( cp <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    cp\shape_aspect_relationship.name = 'connected points')) = 1;
END_ENTITY; -- layer_connection_point_link
(*

```

Formal propositions:

WR1: The related point shall not be the relating point.

WR2: The layer_connection_point_link shall relate to only a layer_connection_point in the related_shape_aspect.

WR3: The layer_connection_point_link shall relate to only a layer_connection_point in the relating_shape_aspect.

WR4: The types of layer_connection_point_link shall be only layer_connection_point_link, shape_aspect, and shape_aspect_relationship.

WR5: The name attribute of the shape_aspect supertype shall be an empty string.

WR6: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

WR7: Each layer_connection_point_link shall be referenced exactly once by a shape_aspect_relationship with a name of 'connected points' by the related_shape_aspect attribute.

5.2.3.1.192 layout_macro_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A layout_macro_component is a printed_component that implements the ARM concept of Layout_macro_component (see 4.2.504) .

EXPRESS specification:

```

*)
ENTITY layout_macro_component
  SUBTYPE OF (printed_component);
WHERE

  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pt_occ <* QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    sa\shape_aspect.description = 'layout macro component shape aspect') |
    SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (pt_occ,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated template') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN
    TYPEOF (it.relatng_shape_aspect))
    )) = 1)) = 1))) = 0;

  WR2: SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'design definition') |
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYOUT_MACRO_DEFINITION'] *
    TYPEOF (ip.relatng_product_definition)) = 1)
    )) = 1;

  WR3: SELF\product_definition.description <> 'printed connector component';
END_ENTITY; -- layout_macro_component
(

```

Formal propositions:

WR1: The layout_macro_component shall have exactly one shape_aspect with a name of 'layout macro component shape aspect' that is related to a layout_macro_floor_plan_template_definition.

WR2: The layout_macro_component shall be associated to exactly one layout_macro_definition as a 'design definition'.

WR3: The layout_macro_component shall not have a description of 'printed connector component'.

5.2.3.1.193 layout_macro_definition

An layout_macro_definition is an interconnect_definition that implements the ARM concept of Layout_macro_definition (see 4.2.505).

EXPRESS specification:

```
*)
ENTITY layout_macro_definition
  SUBTYPE OF (interconnect_definition);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: SIZEOF (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' +
    'RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'design usage') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERCONNECT_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_INTERCONNECT_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_INTERCONNECT_DEFINITION'] *
    TYPEOF (du.relying_product_definition)) = 1) AND
    (du.relying_product_definition.frame_of_reference.name =
    'physical design usage') AND
    (du.relying_product_definition\product_definition.name = 'interconnect module')
    )) = 0;

  WR3: SELF.frame_of_reference\application_context_element.name = 'physical design';
```



```

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pt_occ <* QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    sa\shape_aspect.description = 'layout macro definition shape aspect') |
    SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (pt_occ,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'floor layout view') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE' IN
    TYPEOF (it.relateing_shape_aspect))
    )) = 1)) = 1))) = 0;
END_ENTITY; -- layout_macro_definition
(*

```

Formal propositions:

WR1: The name shall exist.

WR2: The layout_macro_definition shall relate to exactly zero interconnect_definition, externally_defined_interconnect_definition, or library_defined_interconnect_definition that is a physical design usage with a name of 'interconnect module' as the design usage.

WR3: The layout_macro_definition shall be a physical design.

WR4: The layout_macro_definition shall be related to exactly one layout_macro_floor_plan_template.

5.2.3.1.194 layout_macro_definition_terminal_to_usage_terminal_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A layout_macro_definition_terminal_to_usage_terminal_assignment is a shape_aspect and shape_aspect_relationship that implements the ARM concept of Layout_macro_definition_terminal_to_usage_terminal_assignment (see 4.2.506).

EXPRESS specification:

```

*)
ENTITY layout_macro_definition_terminal_to_usage_terminal_assignment
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
UNIQUE
  UR1: SELF\shape_aspect_relationship.name;
WHERE

```

```

WR1: sizeof(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect_relationship ||
    SELF\layout_macro_definition_terminal_to_usage_terminal_assignment ||
    SELF\shape_aspect)) = 0;

WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL' IN
    TYPEOF(SELF\shape_aspect_relationship.relate_shape_aspect);

WR3: SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description
    IN ['component termination passage join terminal',
        'conductive interconnect element terminal',
        'land join terminal',
        'non functional land join terminal',
        'printed component join terminal',
        'via terminal'];

WR4: sizeof(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')) <= 1;

END_ENTITY; -- layout_macro_definition_terminal_to_usage_terminal_assignment
(*)

```

Formal propositions:

UR1: The name shall be unique.

WR1: A layout_macro_definition_terminal_to_usage_terminal_assignment shall not be any other type of shape_aspect_relationship or any other type of shape_aspect.

WR2: A layout_macro_definition_terminal_to_usage_terminal_assignment shall relate to a shape_aspect with a description of 'printed part template terminal'.

WR3: A layout_macro_definition_terminal_to_usage_terminal_assignment shall relate to a shape_aspect with a description listed.

WR4: A layout_macro_definition_terminal_to_usage_terminal_assignment shall be referenced no more than once.

5.2.3.1.195 layout_macro_floor_plan_template

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A layout_macro_floor_plan_template is a part_template_plan_template that implements the ARM concepts of Layout_macro_floor_plan_template (see 4.2.507).

EXPRESS specification:

```

*)
ENTITY layout_macro_floor_plan_template
  SUBTYPE OF (part_template_definition);
WHERE

  WR1: NOT(SELF\shape_aspect.description IN
    ['component termination passage template',
     'component termination passage template',
     'default trace template',
     'inter stratum feature template',
     'printed connector template',
     'printed part cross section template',
     'printed part template',
     'trace template',
     'unsupported passage template',
     'via template',
     'non conductive cross section template']);

  WR2: SIZEOF(QUERY( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ((sar\shape_aspect_relationship.name = 'access maps') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYOUT_MACRO_DEFINITION_TERMINAL_TO_USAGE_TERMINAL_ASSIGNMENT' IN
    TYPEOF (sar.relateing_shape_aspect)))
    )) >=1;
END_ENTITY; -- layout_macro_floor_plan_template
(

```

Formal propositions:

WR1: The layout_macro_floor_plan_template description shall not be any of: 'component termination passage template', 'component termination passage template', 'default trace template', 'inter stratum feature template', 'printed connector template', 'printed part cross section template', 'printed part template', 'trace template', 'unsupported passage template', 'via template', 'non conductive cross section template'.

WR2: The layout_macro_floor_plan_template shall be related to at least one layout_macro_definition-terminal_to_usage_terminal_assignment.

5.2.3.1.196 length_trimmed_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A length_trimmed_terminal is an altered_package_terminal that implements the ARM concept of Length_trimmed_terminal (see 4.2.512).

EXPRESS specification:

```

*)
ENTITY length_trimmed_terminal
  SUBTYPE OF (altered_package_terminal);
WHERE

WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  SIZEOF (
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2)) = 2)) = 1))) = 0);

WR2: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  (SIZEOF (
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2) AND
  (it\representation_item.name =
    'maximum trimmed length')) = 1)) = 1))) = 0);

```

```

WR3: (SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
'minimum trimmed length')) = 1
)) = 1))) = 0);

END_ENTITY;
(*

```

Formal propositions:

WR1: The length_trimmed_terminal shall have a representation that contains exactly two length_measure_with_units.

WR2: The length_trimmed_terminal shall have a representation that contains exactly one length_measure_with_unit with a name of 'maximum trimmed length'.

WR3: The length_trimmed_terminal shall have a representation that contains exactly one length_measure_with_unit with a name of 'minimum trimmed length'.

5.2.3.1.197 library_defined_assembly_definition

A library_defined_assembly_definition is a library_defined_physical_unit that implements the ARM concept of Pca_usage_view (see 4.2.641) in conjunction with Library_item (see ??).

EXPRESS specification:

```

*)
ENTITY library_defined_assembly_definition
  SUBTYPE OF (library_defined_physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: SELF\product_definition.name = 'assembly module';

```

```

WR3: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_MODULE_TERMINAL' IN
      TYPEOF (sa)) AND
      (sa\shape_aspect.description = 'pca terminal')))) >= 2))) = 0);
END_ENTITY; -- library_defined_assembly_definition
(*

```

Formal propositions:

WR1: The name shall exist.

WR2: The library_defined_assembly_definition shall be a definition of a product that has a name of 'assembly module'.

WR3: If the library_defined_assembly_definition has a frame_of_reference name of 'physical design usage', it shall have at least two shape_aspects that are assembly_module_terminals with a description of 'pca terminal'.

5.2.3.1.198 library_defined_bare_die

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A library_defined_bare_die is a library_defined_physical_unit that implements the ARM concept of Bare_die (see 4.2.59) in conjunction with Library_item (see ??).

EXPRESS specification:

```

*)
ENTITY library_defined_bare_die
  SUBTYPE OF (library_defined_physical_unit);
WHERE

```

```

WR1: sizeof (QUERY (ifdu <* QUERY (pdr <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRODUCT_DEFINITION_RELATIONSHIP.' +
' RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'implemented function') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' FUNCTIONAL_UNIT' IN
typeof (ifdu.relying_product_definition)) AND
(ifdu.relying_product_definition.frame_of_reference.name =
'functional design usage')))) = 1;

WR2: sizeof (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
sizeof (QUERY (dut <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'device unit technology') |
dut.relying_property_definition\property_definition.name =
'unit technology')) = 1)) = 1;

WR3: sizeof (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (sizeof (QUERY (sa <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT.OF_SHAPE') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' BARE_DIE_TERMINAL' IN
typeof (sa))) >= 2))) = 0;
END_ENTITY; -- library_defined_bare_die
( *

```

Formal propositions:

WR1: The library_defined_bare_die shall be related to exactly one functional_unit with a frame_of_-reference name of 'functional design usage' as the 'implemented function'.

WR2: The library_defined_bare_die shall have exactly one property_definition that is related to exactly one property_definition with a name of 'unit technology' as the 'device unit technology'.

WR3: The library_defined_bare_die shall have at least two shape_aspects that are bare_die_terminals.

5.2.3.1.199 library_defined_functional_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `library_defined_functional_unit` is an `externally_defined_functional_unit` that implements the ARM concepts of:

- `Ee_product_definition` (see 4.2.323);
- `Functional_unit_definition` (see 4.2.401);
- `Functional_unit_usage_view` (see 4.2.410);
- `Library_ee_product_definition` (see 4.2.513).

EXPRESS specification:

```
*)
ENTITY library_defined_functional_unit
  SUBTYPE OF (externally_defined_functional_unit);
DERIVE
  library : external_source := SELF\externally_defined_item.source;
END_ENTITY; -- library_defined_functional_unit
(*
```

5.2.3.1.200 library_defined_interconnect_definition

A `library_defined_interconnect_definition` is a `library_defined_physical_unit` that implements the ARM concept of `Pcb_usage_view` (see 4.2.643) in conjunction with `Library_item` (see ??).

EXPRESS specification:

```
*)
ENTITY library_defined_interconnect_definition
  SUBTYPE OF (library_defined_physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: SELF\product_definition.name = 'interconnect module';
END_ENTITY; -- library_defined_interconnect_definition
(*
```

Formal propositions:

WR1: The name shall exist.

WR2: The `library_defined_interconnect_definition` shall be a definition with a name of 'interconnect module'.

5.2.3.1.201 library_defined_package

A library_defined_package is a library_defined_physical_unit that implements the ARM concepts of Altered_package (see 4.2.8) and Package (see 4.2.573) in conjunction with Library_item (see ??).

EXPRESS specification:

```

*)
ENTITY library_defined_package
  SUBTYPE OF (library_defined_physical_unit);
WHERE

  WR1: SELF.frame_of_reference.name = 'physical design usage';

  WR2: SIZEOF(USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')) > 0;

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY(sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE' IN
    TYPEOF (sa))) = 1))) = 1;

  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY' IN
    TYPEOF (sa))) = 1))) <= 1;

```

```

WR5: NOT(EXISTS(SELF\product_definition.description) AND
  (SELF\product_definition.description = 'altered package')) OR
  (SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    (pdr\product_definition_relationship.name = 'package preparation') AND
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGE'] *
    TYPEOF (pdr.relateing_product_definition)) = 1))) = 1));
END_ENTITY; -- library_defined_package
(*

```

Formal propositions:

WR1: The library_defined_package shall have a frame_of_reference name of 'physical design usage'.

WR2: The library_defined_package shall have at least one property_definition.

WR3: The library_defined_package shall have one seating_plane.

WR4: The library_defined_package shall have at most one package_body.

WR5: If the library_defined_package has a description of 'altered package', it shall be related to exactly one package, externally_defined_package, or library_defined_package as the 'package preparation'.

Informal propositions:

IP1: Each three-dimensional shape representation of the library_defined_package shall have exactly one geometric_representation_item that is an axis2_placement_3d and that axis2_placement_3d shall define the location and orientation of the seating plane in that representation. The normal of the seating plane in that representation shall be congruent with and the same direction as the Z axis of the axis2_placement_3d. The plane of the seating plane in that representation shall be congruent with the plane established by the X and Y axes of the axis2_placement_3d.

5.2.3.1.202 library_defined_packaged_connector

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A library_defined_packaged_connector is a library_defined_packaged_part that implements the ARM concept of Packaged_connector (see 4.2.587) in conjunction with Library_item (see ??).

EXPRESS specification:

```

*)
ENTITY library_defined_packaged_connector
  SUBTYPE OF (library_defined_packaged_part);
END_ENTITY; -- library_defined_packaged_connector
( *

```

Informal propositions:

IP1: There shall be at least one shape_aspect that is a packaged_connector_terminal_relationship for the library_defined_packaged_part.

5.2.3.1.203 library_defined_packaged_part

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A library_defined_packaged_part is a library_defined_physical_unit that implements the ARM concepts of Altered_packaged_part (see 4.2.10) and Packaged_part (see 4.2.591) in conjunction with Library_item (see ??).

EXPRESS specification:

```

*)
ENTITY library_defined_packaged_part
  SUBTYPE OF (library_defined_physical_unit);
WHERE

```

```

WR1: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART_TERMINAL' IN
        TYPEOF (sa))) >= 2))) = 0));

```

```

WR2: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
      (SIZEOF (USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) = 1);

```

```

WR3: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
(SIZEOF (QUERY (ifu <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'implemented function') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT' IN
TYPEOF (ifu.relying_product_definition)) AND
(ifu.relying_product_definition.frame_of_reference.name =
'functional design usage')))) = 1);

WR4: (NOT (SELF.frame_of_reference.name = 'physical design usage')) OR
(SIZEOF (QUERY (upkg <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'used package') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGE' IN
TYPEOF (upkg.related_product_definition))) = 1);

WR5: (NOT (SELF\product_definition.description = 'altered packaged part')) OR
(SIZEOF (QUERY (bpp <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'base packaged part') |
(SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] *
TYPEOF (bpp.relying_product_definition)) = 1) AND
(bpp.relying_product_definition.frame_of_reference.name =
'physical design usage')))) >= 1);

WR6: (NOT (SELF\product_definition.description = 'altered packaged part')) OR
(SIZEOF (QUERY (upkg <* QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'used package') |
SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] *
TYPEOF (upkg.relying_product_definition)) = 1)) >= 1);
END_ENTITY; -- library_defined_packaged_part
(

```

Formal propositions:

WR1: If the `library_defined_packaged_part` has a `frame_of_reference` name of 'physical design usage', it shall have at least one `shape_aspect` that is a `packaged_part_terminal`.

WR2: If the `library_defined_packaged_part` has a `frame_of_reference` name of 'physical design usage', it shall have exactly one `material_designation`.

WR3: If the `library_defined_packaged_part` has a `frame_of_reference` name of 'physical design usage', it shall be related to exactly one `functional_unit` with a `frame_of_reference` name of 'functional design usage' as the 'implemented function'.

WR4: If the `library_defined_packaged_part` has a `frame_of_reference` name of 'physical design usage', it shall relate to exactly one package to specify the 'used package'.

WR5: If the `library_defined_packaged_part` has a description of 'altered packaged part', it shall be related to at least one `packaged_part`, `externally_defined_packaged_part`, or `library_defined_packaged_part` with a `frame_of_reference` name of 'physical design usage' as the 'base packaged part'.

WR6: If the `library_defined_packaged_part` has a description of 'altered packaged part', it shall be related to at least one package, `externally_defined_package`, or `library_defined_package` as the 'used package'.

5.2.3.1.204 `library_defined_physical_unit`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `library_defined_physical_unit` is an `externally_defined_physical_unit` that implements the ARM concepts of:

- `Assembly_module_usage_view` (see 4.2.51);
- `Ee_product_definition` (see 4.2.323);
- `Interconnect_module_usage_view` (see 4.2.465);
- `Library_ee_product_definition` (see 4.2.513);
- `Physical_unit` (see 4.2.664);
- `Physical_unit_usage_view` (see 4.2.672).

EXPRESS specification:

```

*)
ENTITY library_defined_physical_unit
  SUPERTYPE OF (ONEOF (library_defined_package,
                        library_defined_assembly_definition,
                        library_defined_interconnect_definition,
                        library_defined_packaged_part,
                        library_defined_bare_die))
  SUBTYPE OF (externally_defined_physical_unit);
DERIVE
  library : external_source := SELF\externally_defined_item.source;
END_ENTITY; -- library_defined_physical_unit
( *

```

5.2.3.1.205 library_defined_product_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A library_defined_product_definition is an externally_defined_product_definition that implements the ARM concept of: Library_defined_ee_product_definition (see ??)

EXPRESS specification:

```

*)
ENTITY library_defined_product_definition
  SUBTYPE OF (externally_defined_product_definition);
DERIVE
  library : external_source := SELF\externally_defined_item.source;
END_ENTITY; -- library_defined_product_definition
( *

```

5.2.3.1.206 linear_array_component_definition_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A linear_array_component_definition_link is a product_definition and a product_definition_relationship that implements the ARM concept of Linear_array_placement_group_link (see ??)

EXPRESS specification:

```

*)
ENTITY linear_array_component_definition_link
  SUBTYPE OF (product_definition, product_definition_relationship);
WHERE

```

```

WR1: SELF\product_definition_relationship.related_product_definition :<>:
      SELF\product_definition_relationship.relateing_product_definition;

WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_DEFINITION' IN
      TYPEOF (SELF\product_definition_relationship.related_product_definition);

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_DEFINITION' IN
      TYPEOF (SELF\product_definition_relationship.relateing_product_definition);

WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\product_definition ||
      SELF\product_definition_relationship ||
      SELF\linear_array_component_definition_link)) = 0;

WR5: SELF\product_definition.name = '';

WR6: SELF\product_definition_relationship.name = '';

WR7: SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')) = 1;
END_ENTITY; -- linear_array_component_definition_link
( *
```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The linear_array_component_definition_link shall relate to only a linear_array_placement_group_component_definition in the related_product_definition.

WR3: The linear_array_component_definition_link shall relate to only a linear_array_placement_group_component_definition in the relating_product_definition.

WR4: The types of linear_array_component_definition_link shall be only linear_array_component_definition_link, product_definition, and product_definition_relationship.

WR5: The name attribute of the product_definition supertype shall be an empty string.

WR6: The name attribute of the product_definition_relationship supertype shall be an empty string.

WR7: Each linear_array_component_definition_link shall be referenced exactly once by a product_definition_relationship related_product_definition.

5.2.3.1.207 linear_array_component_shape_aspect_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `linear_array_component_shape_aspect_link` is a `shape_aspect` and a `shape_aspect_relationship` that implements the ARM concept of `Linear_array_placement_group_link` (see ??)

EXPRESS specification:

```

*)
ENTITY linear_array_component_shape_aspect_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_ARRAY_PLACEMENT_GROUP_COMPONENT_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\linear_array_component_shape_aspect_link)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';

  WR7: SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')) = 1;
END_ENTITY; -- linear_array_component_shape_aspect_link
(

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The `linear_array_component_shape_aspect_link` shall relate to only a `linear_array_placement_group_component_shape_aspect` in the `related_shape_aspect`.

WR3: The `linear_array_component_shape_aspect_link` shall relate to only a `linear_array_placement_group_component_shape_aspect` in the `relating_shape_aspect`.

WR4: The types of `linear_array_component_shape_aspect_link` shall be only `linear_array_component_shape_aspect_link`, `shape_aspect`, and `shape_aspect_relationship`.

WR5: The name attribute of the `shape_aspect` supertype shall be an empty string.

WR6: The name attribute of the `shape_aspect_relationship` supertype shall be an empty string.

WR7: Each `linear_array_component_shape_aspect_link` shall be referenced exactly once by a `shape_aspect_relationship` related `shape_aspect`.

5.2.3.1.208 `linear_array_placement_group_component_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `linear_array_placement_group_component_definition` is a `array_placement_group_component_definition` that implements the ARM concept of `Linear_array_placement_group` (see ??) .

EXPRESS specification:

```
*)
ENTITY linear_array_placement_group_component_definition
  SUBTYPE OF (array_placement_group_component_definition);
END_ENTITY; -- linear_array_placement_group_component_definition
(*
```

5.2.3.1.209 `linear_array_placement_group_component_shape_aspect`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `linear_array_placement_group_component_shape_aspect` is a `array_placement_group_component_shape_aspect` that implements the ARM concepts of `Linear_array_placement_group` (see ??) .

EXPRESS specification:

```
*)
ENTITY linear_array_placement_group_component_shape_aspect
  SUBTYPE OF (array_placement_group_component_shape_aspect);
END_ENTITY; -- linear_array_placement_group_component_shape_aspect
(*
```

5.2.3.1.210 `linear_composite_array_shape_aspect`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `linear_composite_array_shape_aspect` is a `composite_shape_aspect` that implements the ARM concept of `Part_linear_array_feature` (see 4.2.615) .

EXPRESS specification:

```

*)
ENTITY linear_composite_array_shape_aspect
  SUBTYPE OF (composite_array_shape_aspect);
END_ENTITY; -- linear_composite_array_shape_aspect
(

```

5.2.3.1.211 linear_composite_array_shape_aspect_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A linear_composite_array_shape_aspect_link is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Part_linear_array_feature_link (see 4.2.616)

EXPRESS specification:

```

*)
ENTITY linear_composite_array_shape_aspect_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relate_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_COMPOSITE_ARRAY_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_COMPOSITE_ARRAY_SHAPE_ASPECT' IN
        TYPEOF (SELF\shape_aspect_relationship.relate_shape_aspect);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\linear_composite_array_shape_aspect_link)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';

  WR7: SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')) = 1;
END_ENTITY; -- linear_composite_array_shape_aspect_link
(

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The linear_composite_array_shape_aspect_link shall relate to only a linear_composite_array_shape_aspect_shape_aspect in the related_shape_aspect.

WR3: The linear_composite_array_shape_aspect_link shall relate to only a linear_composite_array_shape_aspect_shape_aspect in the relating_shape_aspect.

WR4: The types of linear_composite_array_shape_aspect_link shall be only linear_composite_array_shape_aspect_link, shape_aspect, and shape_aspect_relationship.

WR5: The name attribute of the shape_aspect supertype shall be an empty string.

WR6: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

WR7: Each linear_composite_array_shape_aspect_link shall be referenced exactly once by a shape_aspect_relationship related_shape_aspect.

5.2.3.1.212 linear_profile_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A linear_profile_tolerance is a physical_unit_geometric_tolerance that implements the ARM concept of:

- Composite_linear_profile_locating_tolerance (see 4.2.160);
- Composite_linear_profile_refinement_tolerance (see 4.2.161);
- Profile_of_any_line_tolerance (see 4.2.744);
- Profile_of_any_line_tolerance_with_referenced_datum_system (see 4.2.745);
- Profile_of_any_line_tolerance_without_referenced_datum_system (see 4.2.746).

EXPRESS specification:

```

*)
ENTITY linear_profile_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
WHERE

```

```

WR1: ( NOT( SELF\geometric_tolerance.name =
    'linear profile refinement')) OR
    ( SIZEOF( QUERY( gtr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINEAR_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.relativing_geometric_tolerance))
    AND (gtr\geometric_tolerance_relationship.name = 'linear profile refining control')
    AND ((gtr.relativing_geometric_tolerance\geometric_tolerance.name =
    'linear profile locating')
    OR (gtr.relativing_geometric_tolerance\geometric_tolerance.name =
    'linear profile refinement'))))
    )) = 1);

WR2: ( NOT(SELF\geometric_tolerance.name = 'linear profile locating')) OR
    ( SIZEOF( QUERY( gtr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINEAR_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.relativing_geometric_tolerance))
    AND (gtr.relativing_geometric_tolerance\geometric_tolerance.name =
    'linear profile refinement'))
    AND (gtr\geometric_tolerance_relationship.name = 'linear profile refining control'))

WR3: ( NOT( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM' IN TYPEOF(SELF))
    AND ( SELF\geometric_tolerance.name =
    'linear profile refinement')))) OR
    ( SIZEOF( QUERY( gtr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINEAR_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.relativing_geometric_tolerance))
    AND (gtr.relativing_geometric_tolerance\geometric_tolerance.name =
    'linear profile refinement')
    AND (gtr\geometric_tolerance_relationship.name = 'linear profile refining control'))
    )) = 0);

```

```

WR4: ( NOT( SELF\geometric_tolerance.name =
      'linear profile refinement')) OR
      ( SIZEOF (QUERY( gtr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LINEAR_PROFILE_TOLERANCE' IN
      TYPEOF(gtr.relating_geometric_tolerance))
      AND (gtr.related_geometric_tolerance\geometric_tolerance.name =
      'linear profile refinement')
      AND (gtr\geometric_tolerance_relationship.name = 'linear profile refining control')
      )) <= 1);

WR5: NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF));
END_ENTITY; -- linear_profile_tolerance
(*)

```

Formal propositions:

WR1: If the linear_profile_tolerance has a name of 'linear profile refinement', it shall be related as the 'linear profile refining control' to exactly one linear_profile_tolerance with a name of either 'linear profile locating' or 'linear profile refinement'.

WR2: If the linear_profile_tolerance has a name of 'linear profile locating', it shall relate to exactly one linear_profile_tolerance with a name of 'linear profile refinement' to specify the 'linear profile refining control'.

WR3: If the linear_profile_tolerance is not a geometric_tolerance_with_specified_datum_system and has a name of 'linear profile refinement', it shall relate to exactly zero linear_profile_tolerance with a name of 'linear profile refinement' to specify the 'linear profile refining control'.

WR4: If the linear_profile_tolerance has a name of 'linear profile refinement', it shall relate to at most one linear_profile_tolerance with a name of 'linear profile refinement' to specify the 'linear profile refining control'.

WR5: The linear_profile_tolerance shall not be a modified_geometric_tolerance.

Informal propositions:

IP1: If linear_profile_tolerance is a geometric_tolerance_with_specified_datum_system it shall have a name of either 'linear profile', 'linear profile refinement', or 'linear profile locating'; otherwise, it shall have the name of either 'linear profile' or 'linear profile refinement'.

IP2: If the linear_profile_tolerance has a name of 'linear profile refinement', it shall be related as the 'linear profile refining control' to exactly one linear_profile_tolerance with a name of either 'linear profile locating' or 'linear profile refinement'. Both of these linear_profile_tolerances shall specify the same shape_aspect as the toleranced_shape_aspect.

IP3: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and it is related as the 'linear profile refining control' to exactly one `linear_profile_tolerance` with a name of 'linear profile refinement', it shall either not reference a `datum_system` or it shall have less `physical_unit_datum` in its referenced `datum_system` than the `datum_system` referenced by the `linear_profile_tolerance` to which it is related as the 'linear profile refining control'.

IP4: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and it is related as the 'linear profile refining control' to exactly one `linear_profile_tolerance` with a name of 'linear profile locating', it shall either not reference a `datum_system` or it shall have less than or an equal number of `physical_unit_datum` in its referenced `datum_system` than the `datum_system` referenced by the `linear_profile_tolerance` to which it is related as the 'linear profile refining control'.

IP5: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and it is a `geometric_tolerance_with_specified_datum_system`, the `physical_unit_datum` of its referenced `datum_system` shall be `physical_unit_datum` of the `datum_system` of the `linear_profile_tolerance` to which it is related as the 'linear profile refining control' and shall be assigned the same precedence.

IP6: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and it is a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition', the `linear_profile_tolerance` to which it is related as the 'linear profile refining control' shall also be a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition'.

IP7: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and it is a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', the `linear_profile_tolerance` to which it is related as the 'linear profile refining control' shall also be a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', and each of these `property_definition` shall relate to `property_definition` with a description of 'datum feature material condition property' to specify their 'applied material condition property'. These `property_definition` shall be applied to `physical_unit_datum_features` in the context of the `datum_systems` referenced by each of the `linear_profile_tolerance`. The name of the `property_definition` applied to each of the `physical_unit_datum_feature` in the context of the `datum_system` referenced by this `linear_profile_tolerance` shall be the same as the name of the `property_definition` applied to the same `physical_unit_datum_feature` in the context of the `datum_system` referenced by the `linear_profile_tolerance` to which this `linear_profile_tolerance` is related as the 'linear profile refining control'.

IP8: If the `linear_profile_tolerance` has a name of 'linear profile refinement', it shall specify a `measure_with_unit` as its magnitude that is less than the `measure_with_unit` specified as the magnitude of the `linear_profile_tolerance` to which it is related as the 'linear profile refining control'.

IP9: If the `linear_profile_tolerance` has a name of 'linear profile refinement', there shall exist a `property_definition` that has a description of 'true profile' that is related as the 'basic profile' to a `property_definition` of the `tolerance_zone_boundary` of this `linear_profile_tolerance` and to a `property_definition` of the `tolerance_zone_boundary` of the `linear_profile_tolerance` that this `linear_profile_tolerance` is related to as the 'linear profile refining control', and these `property_definition` shall have the description of 'boundary zone definition with specified size' and a name of either 'profile boundary' or 'profile boundary with offsets'.

IP10: If the `linear_profile_tolerance` has a name of 'linear profile refinement' and its `tolerance_zone` has a `property_definition` that has a description of 'tolerance zone orientation' that relates to a `property_definition` with a description of 'viewing plane based orientation' to specify the 'zone orientation', it shall be related to a `linear_profile_tolerance` as the 'linear profile refining control' the `tolerance_zone` of which has a `property_definition` that has a description of 'tolerance zone orientation' that relates to the same `property_definition` with a description of 'viewing plane based orientation' to specify its 'zone orientation'.

IP11: The `toleranced_shape_aspect` of the `linear_profile_tolerance` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP12: The `tolerance_zone_boundary` of the `linear_profile_tolerance` shall have a `property_definition` with a description of 'boundary zone definition with specified size' and a name of either 'profile boundary' or 'profile boundary with offsets'.

IP13: If the `linear_profile_tolerance` is not a `geometric_tolerance_with_specified_datum_system`, its `tolerance_zone` shall have a `property_definition` that has a description of 'tolerance zone orientation' that relates to a `property_definition` with a description of 'viewing plane based orientation' to specify the 'zone orientation'.

5.2.3.1.213 luminous_flux_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `luminous_flux_measure_with_unit` is a `measure_with_unit` that implements the ARM concept of `Luminous_flux_measure_with_unit` (see ??).

EXPRESS specification:

```
*)
ENTITY luminous_flux_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\luminous_flux_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = lumen;
```

```

    WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
        TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- luminous_flux_measure_with_unit
(*)

```

Formal propositions:

WR1: The luminous_flux_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The luminous_flux_measure_with_unit shall reference an si_unit with a name of lumen.

WR3: The unit_component shall be an si_unit.

5.2.3.1.214 magnetic_flux_density_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A magnetic_flux_density_measure_with_unit is a measure_with_unit that implements the ARM concept of Magnetic_flux_density_measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY magnetic_flux_density_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

    WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
        SELF\magnetic_flux_density_measure_with_unit))) = 0;

    WR2: SELF\measure_with_unit.unit_component\si_unit.name = tesla;

    WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
        TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- magnetic_flux_density_measure_with_unit
(*)

```

Formal propositions:

WR1: The magnetic_flux_density_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The magnetic_flux_density_measure_with_unit shall reference an si_unit with a name of tesla.

WR3: The unit_component shall be an si_unit.

5.2.3.1.215 magnetic_flux_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A magnetic_flux_measure_with_unit is a measure_with_unit that implements the ARM concept of Magnetic_flux_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY magnetic_flux_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\magnetic_flux_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = weber;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- magnetic_flux_measure_with_unit
(*
```

Formal propositions:

WR1: The magnetic_flux_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The magnetic_flux_measure_with_unit shall reference an si_unit with a name of weber.

WR3: The unit_component shall be an si_unit.

5.2.3.1.216 make_from_connectivity_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A make_from_connectivity_relationship is a shape_aspect_relationship that implements the ARM concept of Physical_connectivity_make_from_relationship (see 4.2.656).

EXPRESS specification:

```

*)
ENTITY make_from_connectivity_relationship
  SUBTYPE OF (shape_aspect_relationship);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_CONNECTIVITY_DEFINITION'
        IN TYPEOF (SELF.relateing_shape_aspect);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_CONNECTIVITY_DEFINITION'
        IN TYPEOF (SELF.related_shape_aspect);
END_ENTITY; -- make_from_connectivity_relationship
(

```

Formal propositions:

WR1: The make_from_connectivity_relationship shall have a relateing_shape_aspect that references a physical_connectivity_definition.

WR2: The make_from_connectivity_relationship shall have a related_shape_aspect that references a physical_connectivity_definition.

5.2.3.1.217 make_from_feature_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A make_from_feature_relationship is a shape_aspect_relationship that implements the ARM concept of Component_feature_make_from_relationship (see 4.2.120).

EXPRESS specification:

```

*)
ENTITY make_from_feature_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

```

```

WR1: sizeof ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_INTERFACE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT'] *
typeof (SELF.relateing_shape_aspect)) = 1;

WR2: sizeof ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_INTERFACE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT'] *
typeof (SELF.related_shape_aspect)) = 1;

WR3: (NOT ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT' IN
typeof (SELF.relateing_shape_aspect))) OR
(SELF.relateing_shape_aspect\shape_aspect.description IN ['component feature',
'component termination passage interface terminal',
'land or non functional land interface terminal',
'printed connector component interface terminal']);

WR4: (NOT ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT' IN
typeof (SELF.related_shape_aspect))) OR
(SELF.related_shape_aspect\shape_aspect.description IN ['component feature',
'component termination passage interface terminal',
'land or non functional land interface terminal',
'printed connector component interface terminal']);
END_ENTITY; -- make_from_feature_relationship
(*)

```

Formal propositions:

WR1: The make_from_feature_relationship shall have a relating_shape_aspect that is a component_definition, component_interface_terminal, or component_shape_aspect.

WR2: The make_from_feature_relationship shall have a related_shape_aspect that is a component_definition, component_interface_terminal, or component_shape_aspect.

WR3: If the make_from_feature_relationship has a relating_shape_aspect of type component_shape_aspect, the description shall be either 'component feature', 'component termination passage interface terminal', 'land or non functional land interface terminal', or 'printed connector component interface terminal'.

WR4: If the `make_from_feature_relationship` has a `related_shape_aspect` of type `component_shape_aspect`, the description shall be either 'component feature', 'component termination passage interface terminal', 'land or non functional land interface terminal', or 'printed connector component interface terminal'.

5.2.3.1.218 `make_from_functional_unit_terminal_definition_relationship`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `make_from_functional_unit_terminal_definition_relationship` is a `shape_aspect` and `shape_aspect_relationship` that implements the ARM concept of `Functional_unit_terminal_definition_make_from_relationship` (see ??).

EXPRESS specification:

```

*)
ENTITY make_from_functional_unit_terminal_definition_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  UNIQUE
    UR1: SELF\shape_aspect_relationship.relate_shape_aspect,
          SELF\shape_aspect_relationship.related_shape_aspect;
  WHERE

    WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                  'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' ] *
                TYPEOF (SELF.relate_shape_aspect)) = 1;

    WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                  'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' ] *
                TYPEOF (SELF.related_shape_aspect)) = 1;

    WR3: SIZEOF (TYPEOF (SELF) - TYPEOF (SELF\shape_aspect ||
          SELF\shape_aspect_relationship ||
          SELF\make_from_functional_unit_terminal_definition_relationship)) = 0;

    WR4: relate_shape_aspect.of_shape :<>:
          related_shape_aspect.of_shape;

    WR5: SELF\shape_aspect.name = '';

    WR6: SELF\shape_aspect.description = '';

    WR7: SELF\shape_aspect.product_definitional = FALSE;

```

```

WR8: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')) = 0;

WR9: SIZEOF (QUERY( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESIGN_MAKE_FROM_RELATIONSHIP') IN TYPEOF
    (sar.of_shape\property_definition.definition))
    )) = 1;

END_ENTITY; -- make_from_functional_unit_terminal_definition_relationship
(*

```

Formal propositions:

UR1: The make_from_functional_unit_terminal_definition_relationship shall have a relating_shape_aspect and related_shape_aspect combination that is unique.

WR1: The make_from_functional_unit_terminal_definition_relationship shall have a relating_shape_aspect that is a functional_unit_terminal_definition.

WR2: The make_from_functional_unit_terminal_definition_relationship shall have a related_shape_aspect that is a functional_unit_terminal_definition.

WR3: The make_from_functional_unit_terminal_definition_relationship shall not be a complex instance.

WR4: The make_from_functional_unit_terminal_definition_relationship relating_shape_aspect and related_shape_aspect shall not reference the same product_definition_shape.

WR5: The make_from_functional_unit_terminal_definition_relationship shape_aspect supertype name shall be blank.

WR6: The make_from_functional_unit_terminal_definition_relationship shape_aspect supertype description shall be blank.

WR7: The make_from_functional_unit_terminal_definition_relationship shall not be product definitional.

WR8: The make_from_functional_unit_terminal_definition_relationship shall not be referenced by a property_definition.

WR9: The make_from_functional_unit_terminal_definition_relationship shall be related to one design_make_from_relationship.

5.2.3.1.219 make_from_model_port_relationship

A make_from_model_port_relationship is a representation and a representation_relationship that implements the ARM concept of Analytical_model_port_make_from_relationship (see 4.2.19).

EXPRESS specification:

```

*)
ENTITY make_from_model_port_relationship
  SUBTYPE OF (representation, representation_relationship);
  UNIQUE
    UR1: SELF\representation_relationship.rep_1,
        SELF\representation_relationship.rep_2;
  WHERE

    WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL_PORT' ] *
        TYPEOF (SELF.rep_1)) = 1;

    WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL_PORT' ] *
        TYPEOF (SELF.rep_2)) = 1;

    WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\representation ||
        SELF\representation_relationship ||
        SELF\make_from_model_port_relationship)) = 0;

    WR4: rep_1 :<>: rep_2;

    WR5: SIZEOF(QUERY( rr1 <* USEDIN(SELF.rep_1,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') |
        (SIZEOF(QUERY(rr2 <* USEDIN(SELF.rep_2,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') |
        (rr1.rep_1 :=: rr2.rep_1)
        )) = 0)))) = 1;

    WR6: SIZEOF (QUERY( rr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') |
        (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL_MAKE_FROM_RELATIONSHIP') IN TYPEOF (rr.rep_1))
        )) = 1;

END_ENTITY; -- make_from_model_port_relationship
(

```

Formal propositions:

UR1: The make_from_model_port_relationship shall have a unique combination of rep_1 and rep_2.

WR1: The make_from_model_port_relationship shall have a rep_1 that is an analytical_model_port.

WR2: The make_from_model_port_relationship shall have a rep_2 that is an analytical_model_port.

WR3: The make_from_model_port_relationship shall not be a complex instance.

WR4: The make_from_model_port_relationship shall not reference the same port.

WR5: The make_from_model_port_relationship shall not reference the same analytical_model.

WR6: The make_from_model_port_relationship shall be related to one analytical_model_make_from_-relationship.

5.2.3.1.220 make_from_part_feature_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A make_from_part_feature_relationship is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Part_feature_make_from_relationship (see 4.2.610).

EXPRESS specification:

```

*)
ENTITY make_from_part_feature_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  UNIQUE
    UR1: SELF\shape_aspect_relationship.relate_shape_aspect,
        SELF\shape_aspect_relationship.related_shape_aspect;
  WHERE

    WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT'] *
        TYPEOF (SELF.relate_shape_aspect)) = 0;

    WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT'] *
        TYPEOF (SELF.related_shape_aspect)) = 0;

    WR3: SIZEOF (TYPEOF (SELF) - TYPEOF (SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\make_from_part_feature_relationship)) = 0;

```

```

WR4: SELF.related_shape_aspect.product_definitional = TRUE;

WR5: SELF.relying_shape_aspect.product_definitional = TRUE;

WR6: SELF\shape_aspect.name = '';

WR7: SELF\shape_aspect.description = '';

WR8: SELF\shape_aspect.product_definitional = FALSE;

WR9: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')) = 0;

WR10: sizeof (QUERY( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESIGN_MAKE_FROM_RELATIONSHIP') IN typeof
    (sar.of_shape\property_definition.definition))
    )) = 1;
END_ENTITY; -- make_from_part_feature_relationship
(*

```

Formal propositions:

UR1: The make_from_model_port_relationship shall have a unique combination of related_shape_aspect and relating_shape_aspect.

WR1: The make_from_part_feature_relationship shall have a relating_shape_aspect that is not a Non_feature_shape_element and that is not a Component_feature.

WR2: The make_from_part_feature_relationship shall have a related_shape_aspect that is not a Non_feature_shape_element and that is not a Component_feature.

WR3: The make_from_part_feature_relationship shall not be a complex instance.

WR4: The make_from_part_feature_relationship related_shape_aspect shall be on the boundary of the part or template.

WR5: The make_from_part_feature_relationship relating_shape_aspect shall be on the boundary of the part or template.

WR6: The make_from_part_feature_relationship shall have a name inherited from the shape_aspect that is blank.

WR7: The make_from_part_feature_relationship shall have a description inherited from the shape_aspect that is blank.

WR8: The make_from_part_feature_relationship shall not be on the boundary of the part or template.

WR9: The make_from_part_feature_relationship shall not be referenced by a property_definition.

WR10: The make_from_part_feature_relationship shall be related to exactly one design_make_from_relationship.

5.2.3.1.221 manifold_constraining_context_dependent_shape_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A manifold_constraining_context_dependent_shape_representation is a context_dependent_shape_representation and a representation_relationship that implements the ARM concept of An_unknown_arm_type (see ??).

EXPRESS specification:

```
*)
ENTITY manifold_constraining_context_dependent_shape_representation
  SUBTYPE OF (context_dependent_shape_representation, representation_relationship);
END_ENTITY; -- manifold_constraining_context_dependent_shape_representation
(*
```

5.2.3.1.222 material_electrical_conductivity_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A material_electrical_conductivity_category is a group that specifies the attribute electrical_conductivity_characteristic for the ARM concept of Material_electrical_conductivity_requirement_occurrence (see 4.2.532) and the electrical_conductivity_characteristic for the ARM concept of Material_electrical_conductivity_requirement_occurrence (see 4.2.532).

EXPRESS specification:

```
*)
ENTITY material_electrical_conductivity_category
  SUBTYPE OF (group);
WHERE

  WR1: SELF\group.name IN ['conductive', 'non conductive', 'resistive',
    'semi conductive'];
END_ENTITY; -- material_electrical_conductivity_category
(*
```

Formal propositions:

WR1: The material_electrical_conductivity_category shall have a name of 'conductive', 'non conductive', 'resistive', or 'semi conductive'

5.2.3.1.223 material_functional_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A material_functional_category is a group that specifies the role of material for the ARM concepts of:

- Marking (see 4.2.526);
- Packaged_part (see 4.2.591);
- Part_device (see 4.2.606).

.

EXPRESS specification:

```
*)  
ENTITY material_functional_category  
  SUBTYPE OF (group);  
WHERE  
  
  WR1: SELF\group.name IN  
    ['bonding agent', 'potting compound', 'marking material'];  
END_ENTITY; -- material_functional_category  
(*
```

Formal propositions:

WR1: The material_functional_category shall have a name of 'bonding agent', 'potting compound', or 'marking material'.

5.2.3.1.224 material_optical_conductivity_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A material_optical_conductivity_category is a group that specifies the optical conductivity characteristic for the ARM concept of Ee_material (see 4.2.318).

EXPRESS specification:

```

*)
ENTITY material_optical_conductivity_category
  SUBTYPE OF (group);
WHERE

  WR1: SELF\group.name IN ['conductive', 'non conductive', 'resistive',
    'semi conductive'];
END_ENTITY; -- material_optical_conductivity_category
( *

```

Formal propositions:

WR1: The material_optical_conductivity_category shall have a name of 'conductive', 'non conductive', 'resistive', or 'semi conductive'

5.2.3.1.225 material_removal_component_shape_aspect

A material_removal_component_shape_aspect is a component_shape_aspect that implements the ARM concept of Material_removal_component (see ??).

EXPRESS specification:

```

*)
ENTITY material_removal_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
END_ENTITY; -- material_removal_component_shape_aspect
( *

```

5.2.3.1.226 material_thermal_conductivity_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A material_thermal_conductivity_category is a group that specifies the thermal conductivity characteristic for the ARM concept of Ee_material (see 4.2.318).

EXPRESS specification:

```

*)
ENTITY material_thermal_conductivity_category
  SUBTYPE OF (group);
WHERE

```

```

    WR1: SELF\group.name IN ['conductive', 'non conductive', 'resistive',
        'semi conductive'];
END_ENTITY; -- material_thermal_conductivity_category
(*)

```

Formal propositions:

WR1: The material_thermal_conductivity_category shall have a name of 'conductive', 'non conductive', 'resistive', or 'semi conductive'

5.2.3.1.227 minimally_defined_bare_die_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A minimally_defined_bare_die_terminal is a shape_aspect that implements the ARM concepts of Guided_wave_bare_die_terminal (see 4.2.426) or Minimally_defined_bare_die_terminal (see 4.2.545).

EXPRESS specification:

```

*)
ENTITY minimally_defined_bare_die_terminal
    SUPERTYPE OF (bare_die_terminal)
    SUBTYPE OF (shape_aspect);
WHERE

    WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE' IN
        TYPEOF (SELF.of_shape.definition);

    WR2: SIZEOF (QUERY (eca <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'external connection area') |
        eca.related_shape_aspect\shape_aspect.description = 'connection zone')) <= 1;

    WR3: SIZEOF (USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1;

```

```

WR4: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'member connected terminal') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION'
    IN TYPEOF (mct.relate_shape_aspect))) <= 1;
END_ENTITY; -- minimally_defined_bare_die_terminal
(*)

```

Formal propositions:

WR1: The `minimally_defined_bare_die_terminal` shall be a terminal of a `bare_die`.

WR2: The `minimally_defined_bare_die_terminal` shall relate to at most one `shape_aspect` that has a description of 'connection zone' as the 'external connection area'.

WR3: The `minimally_defined_bare_die_terminal` shall have at most one material designation.

WR4: The `minimally_defined_bare_die_terminal` shall be related to at most one `part_connected_terminals_definition` as a 'member connected terminal'.

5.2.3.1.228 **model_parameter**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `model_parameter` is a `representation_item` that implements the ARM concept of `Model_parameter` (see 4.2.549).

EXPRESS specification:

```

*)
ENTITY model_parameter
  SUBTYPE OF (representation_item);
WHERE

  WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;

  WR2: SIZEOF (QUERY (cri <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COORDINATED_REPRESENTATION_ITEM' IN TYPEOF(cri)) AND
    (cri\representation.name =
    'model parameter with valid range value')))) <= 1;

```

```

WR3: sizeof (QUERY (aga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PARAMETER_TYPE' IN
    TYPEOF (aga.assigned_group)) AND
    (aga.assigned_group\group.name IN ['string property type',
    'logical property type', 'physical property type',
    'boolean property type']))) = 1;
END_ENTITY; -- model_parameter
(*

```

Formal propositions:

WR1: The model_parameter shall be referenced by exactly one document.

WR2: The model_parameter shall be an item in at most one coordinated_representation_item that has a representation name of 'model parameter with valid range value'.

WR3: The model_parameter shall be assigned to exactly one parameter_type with a name of 'string property type', 'logical property type', 'physical property type', or 'boolean property type'.

5.2.3.1.229 mounting_restriction_area

A mounting_restriction_area is a shape_aspect that implements the ARM concept of Mounting_restriction_area (see 4.2.550).

EXPRESS specification:

```

*)
ENTITY mounting_restriction_area
  SUBTYPE OF (shape_aspect);
WHERE

```

WR1: EXISTS(SELF.of_shape.definition\product_definition.name);

WR2: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_UNIT' IN TYPEOF (SELF.of_shape.definition));

WR3: SELF.of_shape.definition.frame_of_reference.name IN ['physical design'];

WR4: SELF.of_shape.definition\product_definition.name = 'assembly module';

```

WR5: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CSG_2D_SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation)))) = 1))) = 0;

WR6: sizeof (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    ((sar\shape_aspect_relationship.name = 'mounting surface') AND
    (sar.related_shape_aspect\shape_aspect.description =
    'interconnect module component surface feature')))) = 1;

WR7: sizeof (typeof (SELF) - (typeof (SELF\shape_aspect ||
    SELF\mounting_restriction_area))) = 0;

END_ENTITY; -- mounting_restriction_area
(*)

```

Formal propositions:

WR1: The name of the physical_unit shall exist.

WR2: The mounting_restriction_area shall be an aspect of the shape of a physical_unit.

WR3: The mounting_restriction_area shall be an aspect of a product_definition that is in the physical design context.

WR4: The mounting_restriction_area shall be an aspect of the shape of an assembly module.

WR5: The mounting_restriction_area shall have exactly one shape_representation that shall be a csg_2d_shape_representation.

WR6: The mounting_restriction_area shall be related to exactly one component_shape_aspect with a description of 'interconnect module component surface feature' as the 'mounting surface'.

WR7: The mounting_restriction_area shall not be any other type of shape_aspect.

5.2.3.1.230 mounting_restriction_volume

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A mounting_restriction_volume is a shape_aspect that implements the ARM concept of Mounting_restriction_volume (see 4.2.551).

EXPRESS specification:

```

*)
ENTITY mounting_restriction_volume
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT' IN TYPEOF (SELF.of_shape.definition));

  WR2: SELF.of_shape.definition.frame_of_reference.name IN ['physical design'];

  WR3: SELF.of_shape.definition.name = 'assembly module';

  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation)) AND
    (pdr.used_representation\representation.context_of_items\
    geometric_representation_context.coordinate_space_dimension = 3)
    )) = 1))) = 0;

  WR5: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN
    TYPEOF (sar.related_shape_aspect)) AND
    (sar\shape_aspect_relationship.name = 'mounting surface') AND
    (sar.related_shape_aspect\shape_aspect.description =
    'interconnect module component surface feature')))) = 1;

  WR6: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\shape_aspect ||
    SELF\mounting_restriction_volume))) = 0;
END_ENTITY; -- mounting_restriction_volume
(

```


Formal propositions:

WR1: The mounting_restriction_volume shall be an aspect of the shape of a physical_unit.

WR2: The mounting_restriction_volume shall be an aspect of a product_definition that is in the physical design context.

WR3: The mounting_restriction_volume shall be an aspect of the shape of an assembly module.

WR4: The mounting_restriction_volume shall have exactly one shape_representation that shall be a 3 dimensional shape representation.

WR5: The mounting_restriction_volume shall be related to exactly one component_shape_aspect with a description of 'interconnect module component surface feature' as the 'mounting surface'.

WR6: The mounting_restriction_volume shall not be any other type of shape_aspect.

5.2.3.1.231 multi_layer_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A multi_layer_component_shape_aspect is a component_shape_aspect that implements the ARM concept of Multi_layer_stratum_feature_template_component (see 4.2.554).

EXPRESS specification:

```
*)
ENTITY multi_layer_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
END_ENTITY; -- multi_layer_component_shape_aspect
(*)
```

5.2.3.1.232 multi_layer_material_removal_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A multi_layer_material_removal_component_shape_aspect is a material_removal_component_shape_aspect that implements the ARM concept of Multi_layer_material_removal_laminate_component (see 4.2.552).

EXPRESS specification:

```
*)
ENTITY multi_layer_material_removal_component_shape_aspect
  SUBTYPE OF (material_removal_component_shape_aspect);
END_ENTITY; -- multi_layer_material_removal_component_shape_aspect
(*)
```

5.2.3.1.233 multi_layer_stratum_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A multi_layer_stratum_feature is a stratum_feature that implements the ARM concept of Multi_layer_stratum_feature (see 4.2.553).

EXPRESS specification:

```
*)
ENTITY multi_layer_stratum_feature
  SUBTYPE OF (stratum_feature);
END_ENTITY; -- multi_layer_stratum_feature
(*
```

5.2.3.1.234 network_node_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A network_node_definition is a product_definition that implements the ARM concept of Functional_unit_network_node_definition (see 4.2.403).

EXPRESS specification:

```
*)
ENTITY network_node_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: SELF.frame_of_reference.name = 'functional network design';
```

```

WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (funtdna <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name =
    'functional unit network terminal definition node assignment') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (funtdna.related_property_definition)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' IN
    TYPEOF
    (funtdna.related_property_definition.definition)))) <= 1)) <= 1;

WR3: SIZEOF (QUERY (funn <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' IN
    TYPEOF (funn.relying_product_definition)) AND
    (funn.relying_product_definition.frame_of_reference.name =
    'functional network design')) = 1;
END_ENTITY; -- network_node_definition
(*)

```

Formal propositions:

WR1: The network_node_definition shall have its context element with a name of 'functional network design'.

WR2: The network_node_definition shall have at most one property that relates to at most one product_definition_shape of a functional_unit_terminal_definition to specify the 'functional unit network terminal definition node assignment'.

WR3: The network_node_definition shall be related to exactly one functional_unit that has a context element with a name of 'functional network design' as the 'functional unit network node'.

Informal propositions:

IP1: This is an IP for now, as it obviously uses arm constructs. It will be converted to DERIVE and then the electrical and thermal network structural validity can be exchanged based on approval. Probably need something like "valid structure".

NOTE

5.2.3.1.235 opposing_boundary_dimensional_size

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `opposing_boundary_dimensional_size` is a `dimensional_size` that implements the ARM concept of `Dimensional_size_based_on_opposing_boundaries` (see 4.2.302).

EXPRESS specification:

```
*)
ENTITY opposing_boundary_dimensional_size
  SUBTYPE OF (dimensional_size);
WHERE

    WR1: SELF\dimensional_size.name IN ['angular', 'linear'];
END_ENTITY; -- opposing_boundary_dimensional_size
(*
```

Formal propositions:

WR1: The name of the `opposing_boundary_dimensional_size` shall be either 'angular' or 'linear'.

Informal propositions:

IP1: The `opposing_boundary_dimensional_size` applies_to a `composite_unit_shape_aspect`.

5.2.3.1.236 owner_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `owner_assignment` is an `person_and_organization_assignment` that specifies a set of `owner_assigned_item`. An item that is owned cannot be changed without the permission of the owner.

EXAMPLE 1 A component position may be owned by the mechanical design department, in which case movement of the component is to be negotiated with that department.

EXPRESS specification:

```
*)
ENTITY owner_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF owner_assigned_item;
END_ENTITY; -- owner_assignment
(*
```

5.2.3.1.237 package

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package is a `physical_unit` that implements the ARM concepts of `Altered_package` (see 4.2.8) and `Package` (see 4.2.573).

EXPRESS specification:

```

*)
ENTITY package
  SUBTYPE OF (physical_unit);
WHERE

  WR1: SELF.frame_of_reference.name = 'physical design usage';
  -- WR2 deleted

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY' IN
    TYPEOF (sa))) = 1))) <= 1;

  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_TERMINAL' IN
    TYPEOF (sa))) >= 1))) >= 1;

```

```

WR5: (NOT (SELF\product_definition.description = 'altered package')) OR
      (SIZEOF (QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
        (pdr\product_definition_relationship.name = 'package alteration') AND
        (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE'] *
        TYPEOF (pdr.relatng_product_definition)) = 1))) = 1));

WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation\representation.name = 'package mounting data')) = 1))) = 1;

WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (pdr.used_representation\representation.name = 'package mounting data')
        AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | item\representation_item.name =
        'maximum body height above seating plane')) = 1)
        )) = 1))) = 1;

WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (pdr.used_representation\representation.name = 'package mounting data')
        AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | item\representation_item.name =
        'maximum body height below seating plane')) = 1)
        )) = 1))) <= 1;

```

```

WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'maximum body clearance above seating plane')) = 1)
    )) = 1))) <= 1;

```

```

WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'maximum body clearance below seating plane')) = 1)
    )) = 1))) <= 1;

```

```

WR11: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'minimum body clearance above seating plane')) = 1)
    )) = 1))) <= 1;

```

```

WR12: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'minimum body clearance below seating plane')) = 1)
    )) = 1))) <= 1;

```

```

WR13: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'maximum lead length below seating plane')) = 1)
    )) = 1))) <= 1;

```

```

WR14: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (pdr.used_representation\representation.name = 'package mounting data')
    AND (SIZEOF (QUERY (item <* USEDIN( pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | item\representation_item.name =
    'least lead length below seating plane')) = 1)
    )) = 1))) <= 1;

```



```

WR15: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT' IN TYPEOF(sa)) AND
    (SIZEOF (QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    ((sar\shape_aspect_relationship.name = 'package seating plane') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE' IN TYPEOF(sar.related_shape_aspect)))) )) = 1))
    )) = 1)
    )
    ) = 1;

WR16: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE' IN
    TYPEOF (sa)))) = 1))) = 1;

WR17: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT' IN TYPEOF(sa)) AND
    (SIZEOF (QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    ((sar\shape_aspect_relationship.name = 'of datum reference plane') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_REFERENCE_FRAME' IN TYPEOF(sar.related_shape_aspect)))) )) = 1))
    )) = 1)
    )
    ) <= 1;

```

```

WR18: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRIMARY_REFERENCE_TERMINAL' IN TYPEOF(sa))
    )) = 1)
    )
    ) <= 1;
END_ENTITY; -- package
( *

```

Formal propositions:

WR1: The package shall have a frame_of_reference name of 'physical design usage'.

WR2: The package shall have at most one package_body.

WR3: The package shall have at least one package_terminal.

WR4: If the package has a description of 'altered package', it shall be related to exactly one package, externally_defined_package, or library_defined_package as the 'package alteration'.

WR5: The package shall have exactly one representation with the name of 'package mounting data'.

WR6: The package shall have one item in its representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'maximum body height above seating plane'.

WR7: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'maximum body height below seating plane'.

WR8: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'maximum body clearance above seating plane'.

WR9: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'maximum body clearance below seating plane'.

WR10: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'minimum body clearance above seating plane'.

WR11: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'minimum body clearance below seating plane'.

WR12: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'maximum lead length below seating plane'.

WR13: The package shall have no more than one item in the representation with the name of 'package mounting data' that shall be a length_measure_with_unit and that shall have a name of 'least lead length below seating plane'.

WR14: The seating_plane for this package shall be related to a shape_aspect of this package through a shape_aspect_relationship with the name of 'package seating plane'.

WR15: The package shall have exactly one seating plane.

WR16: The package shall have at most one datum_reference_frame.

WR17: The package shall have at most one primary_reference_terminal.

Informal propositions:

IP1: Each three-dimensional shape representation of the package shall have exactly two geometric_representation_item that are axis2_placement_3d and one axis2_placement_3d shall define the location and orientation of the seating plane in that representation. The normal of the seating plane in that representation shall be congruent with and the same direction as the Z axis of the axis2_placement_3d. The plane of the seating plane in that representation shall be congruent with the plane established by the X and Y axes of the axis2_placement_3d.

IP2: Each two-dimensional shape representation of the package shall have exactly one geometric_representation_item that is an axis2_placement_2d that shall define the origin of the geometric model, and the location and orientation of the seating plane.

IP3: The package shall have no more than one representation with the name of 'material property data' that shall be related to cartesian_points with names of 'least material condition centroid location' and 'maximum material condition centroid location'.

5.2.3.1.238 package_body

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body is a shape_aspect that implements the ARM concept of Package_body (see 4.2.574).

EXPRESS specification:

```
* )
ENTITY package_body
  SUBTYPE OF (shape_aspect);
WHERE
```

```

WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'PACKAGE',
   'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'EXTERNALLY_DEFINED_PACKAGE',
   'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'LIBRARY_DEFINED_PACKAGE']) = 1;

WR2: SIZEOF (USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MATERIAL_DESIGNATION.DEFINITIONS')) = 1;
END_ENTITY; -- package_body
( *

```

Formal propositions:

WR1: The package_body shall be the aspect of the shape of a package, externally_defined_package, or library_defined_package.

WR2: The package_body shall have exactly one material_designation.

5.2.3.1.239 package_body_bottom_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body_bottom_surface is a package_body_surface that implements the ARM concept of Package_body_bottom_surface (see 4.2.575).

EXPRESS specification:

```

*)
ENTITY package_body_bottom_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_bottom_surface
( *

```

5.2.3.1.240 package_body_edge_segment_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body_edge_segment_surface is a shape_aspect and shape_aspect_relationship that implements the ARM concept of Package_body_edge_segment_surface (see 4.2.576).

EXPRESS specification:

```

*)
ENTITY package_body_edge_segment_surface
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_SEGMENT_VERTEX' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_SEGMENT_VERTEX' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);

  WR3: SELF\shape_aspect_relationship.relying_shape_aspect :<>:
        SELF\shape_aspect_relationship.related_shape_aspect;

  WR4: SIZEOF (QUERY (ce < * QUERY (sar < * USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'composed surface') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY_EDGE_SURFACE' IN TYPEOF
        (ce.relying_shape_aspect) ))) = 1;

  WR5: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\package_body_edge_segment_surface)) = 0;
END_ENTITY; -- package_body_edge_segment_surface
( *

```

Formal propositions:

WR1: The package_body_edge_segment_surface shall be related to exactly one edge_segment_vertex through a shape_aspect_relationship related attribute.

WR2: The package_body_edge_segment_surface shall be related to exactly one edge_segment_vertex through a shape_aspect_relationship relating attribute.

WR3: The edge_segment_vertex instances referenced shall not be the same instance of edge_segment_vertex.

WR4: The package_body_edge_segment_surface shall be related to exactly one package_body_edge_surface as the 'composed surface'.

WR5: The package_body_edge_segment_surface shall not be in a complex instance other than with it's immediate supertypes.

5.2.3.1.241 package_body_edge_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body_edge_surface is a shape_aspect that implements the ARM concept of Package_body_edge_surface (see 4.2.577).

EXPRESS specification:

```

*)
ENTITY package_body_edge_surface
  SUBTYPE OF (package_body_surface);
WHERE

  WR1: SELF\shape_aspect.product_definitional;

  WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\part_mounting_feature ||
    SELF\package_body_edge_surface ||
    SELF\package_body_surface)) = 0;
END_ENTITY; -- package_body_edge_surface
( *

```

Formal propositions:

WR1: The package_body_edge_surface shall be on the boundary of the product.

WR2: The package_body_edge_surface shall be no other subtype of shape_aspect.

5.2.3.1.242 package_body_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body_surface is a shape_aspect that implements the ARM concept of Package_body_surface (see 4.2.578).

EXPRESS specification:

```

*)
ENTITY package_body_surface
ABSTRACT SUPERTYPE OF (ONEOF(package_body_top_surface,
                               package_body_edge_surface,
                               package_body_bottom_surface))
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'PACKAGE',
     'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'EXTERNALLY_DEFINED_PACKAGE',
     'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'LIBRARY_DEFINED_PACKAGE']) = 1;

  WR2: SELF\shape_aspect.product_definitional;

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (TYPEOF(sar\shape_aspect_relationship.relate_shape_aspect) =
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
     'PACKAGE_BODY']))) = 1;
END_ENTITY; -- package_body_surface
(

```

Formal propositions:

WR1: The package_body_surface shall be the aspect of the shape of a package, externally_defined_package, or library_defined_package.

WR2: The package_body_surface shall be on the boundary of the product.

WR3: The package_body_surface shall be related to exactly one package_body.

5.2.3.1.243 package_body_top_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A package_body_top_surface is a package_body_surface that implements the ARM concept of Package_body_top_surface (see 4.2.579).

EXPRESS specification:

```

*)
ENTITY package_body_top_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_top_surface
( *

```

5.2.3.1.244 package_terminal

A package_terminal is a shape_aspect that implements the ARM concepts of:

- Altered_package_terminal (see 4.2.9);
- Guided_wave_terminal (see 4.2.427);
- Length_trimmed_terminal (see 4.2.512);
- Package_terminal (see 4.2.581);
- Shape_formed_terminal (see 4.2.817);
- Surface_prepped_terminal (see 4.2.881);
- Wire_terminal (see 4.2.963).

EXPRESS specification:

```

*)
ENTITY package_terminal
  SUPERTYPE OF (((altered_package_terminal ANDOR guided_wave_terminal)
    ANDOR wire_terminal) ANDOR primary_reference_terminal)
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation))) >= 1))) = 0;

```



```

WR2: SIZEOF (TYPEOF (SELF.of_shape.definition) *
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'PACKAGE',
   'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'EXTERNALLY_DEFINED_PACKAGE',
   'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
   'LIBRARY_DEFINED_PACKAGE']) = 1;

WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION' IN TYPEOF (sar.relatng_shape_aspect.of_shape.definition)) AND
  (sar\shape_aspect_relationship.name = 'terminal core material') AND
  (sar.relatng_shape_aspect.of_shape.definition.
   frame_of_reference\application_context_element.name = 'material definition')
  )) <= 1;

WR4: SIZEOF (QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION' IN TYPEOF (sar.relatng_shape_aspect.of_shape.definition)) AND
  (sar\shape_aspect_relationship.name = 'terminal surface material') AND
  (sar.relatng_shape_aspect.of_shape.definition.
   frame_of_reference\application_context_element.name = 'material definition')
  )) = 1;

WR5: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'member connected terminal') |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PART_CONNECTED_TERMINALS_DEFINITION'
  IN TYPEOF (mct.relatng_shape_aspect))) <= 1;

WR6: SIZEOF (QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
  (sar\shape_aspect_relationship.name = 'external connection zone') AND
  (sar.related_shape_aspect\shape_aspect.description = 'connection zone')) >= 0;

```

```

WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF (QUERY (lmwu <* QUERY (it <* pdr.used_representation.items |
    SIZEOF (
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) |
    lmwu\representation_item.name = 'maximum terminal diametrical extent'))
    <= 1)) <= 1)) <= 1;

WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF (QUERY (lmwu <* QUERY (it <* pdr.used_representation.items |
    SIZEOF (
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) |
    lmwu\representation_item.name = 'minimum terminal diametrical extent'))
    <= 1)) <= 1)) <= 1;

WR9: SIZEOF(QUERY ( sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'precedent feaure' ))
    <= 1;

WR10: SIZEOF(QUERY ( sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'precedent feaure' ))
    <= 1;

```

```

WR11: SIZEOF (QUERY (eca <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'seating plane zone') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONNECTION_ZONE_INTERFACE_PLANE_RELATIONSHIP' IN
    TYPEOF (eca.related_shape_aspect))) <= 1;
END_ENTITY; -- package_terminal
(*

```

Formal propositions:

WR1: The package_terminal shall have at least one shape_representation.

WR2: The package_terminal shall be the aspect of the shape of a package, externally_defined_package, or library_defined_package.

WR3: The package_terminal shall have no more than one shape_aspect_relationship with a name of 'terminal core material' that relates to a product_definition with a frame of reference of 'material definition'.

WR4: The package_terminal shall have no more than one shape_aspect_relationship with a name of 'terminal core material' that relates to a product_definition with a frame of reference of 'material definition'.

WR5: The package_terminal shall be related to at most one part_connected_terminals_definition as a 'member connected terminal'.

WR6: The package_terminal shall be related to zero or more shape_aspect with a description of 'connection zone' as the 'external connection zone'.

WR7: The package_terminal shall have at most one property_definition that has at most one representation that contains at most one length_measure_with_unit with a name of 'maximum terminal diametrical extent'.

WR8: The package_terminal shall have at most one property_definition that has at most one representation that contains at most one length_measure_with_unit with a name of 'minimum terminal diametrical extent'.

WR9: The package_terminal shall be referenced by no more than one shape_aspect_relationship with a description of 'precedent feature' as the relating_shape_aspect.

WR10: The package_terminal shall be referenced by no more than one shape_aspect_relationship with a description of 'precedent feature' as the related_shape_aspect.

WR11: The package_terminal shall be related to no more than one connection_zone_interface_plane_relationship.

5.2.3.1.245 packaged_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A packaged_component is a component_definition that implements the ARM concepts of:

- Packaged_component (see 4.2.585);
- Packaged_connector_component (see 4.2.588);
- Routed_packaged_component (see ??).

EXPRESS specification:

*)

```
ENTITY packaged_component
  SUBTYPE OF (component_definition);
WHERE
```

```
WR1: SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  pdr\product_definition_relationship.name = 'instantiated part') |
  (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PACKAGED_PART',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'EXTERNALLY_DEFINED_PACKAGED_PART',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LIBRARY_DEFINED_PACKAGED_PART'] *
  TYPEOF (ip.relateing_product_definition)) = 1) AND
  (ip.relateing_product_definition.frame_of_reference.name =
  'physical design usage')) = 1;
```

```
WR2: SIZEOF (QUERY (pa <* QUERY (pdr <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  pdr\product_definition_relationship.name = 'package alternate') |
  (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PACKAGE',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'EXTERNALLY_DEFINED_PACKAGE',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LIBRARY_DEFINED_PACKAGE'] *
  TYPEOF (pa.relateing_product_definition)) = 1) AND
  (pa.relateing_product_definition\product_definition.description =
  'altered package')) <= 1;
```

```

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (sa)) AND
    (sa\shape_aspect.description = 'packaged component join terminal')))) >= 1))) = 0;

WR4: (NOT (SELF\product_definition.description = 'packaged connector component')) OR
    (SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'instantiated part') |
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'] *
    TYPEOF (ip.relateing_product_definition)) = 1) AND
    (TRUE))) = 1);

WR5: (NOT (SELF\product_definition.description = 'packaged connector component')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (sa)) AND
    (sa\shape_aspect.description = 'packaged component join terminal'))))
    >= 1))) = 0);

```

```

WR6: (NOT (SELF\product_definition.description = 'routed packaged component')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      SIZEOF (QUERY (pd <* USEDIN (sa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF (USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'))
      = 1)) = 1)) = 1))) = 0);
END_ENTITY; -- packaged_component
(*

```

Formal propositions:

WR1: The packaged_component shall be related, as the 'instantiated part', to exactly one packaged_part, externally_defined_packaged_part, or library_defined_packaged_part that is in the 'physical design usage' product_definition_context.

WR2: The packaged_component shall be related, as the 'package alternate', to at most one package, externally_defined_package, or library_defined_package that has a description of 'altered package'.

WR3: The packaged_component shall have at least one shape_aspect that is a component_terminal with a description of 'packaged component join terminal'.

WR4: If the packaged_component has description of 'packaged connector component', the packaged_component shall be related, as the 'instantiated part', to exactly one packaged_part, externally_defined_packaged_part, or library_defined_packaged_part that is in the 'packaged connector' product_definition_context.

WR5: If the packaged_component has a description of 'packaged connector component', it shall have at least one shape_aspect that is a component_interface_terminal with a description of 'packaged component join terminal'.

WR6: If the packaged_component has a description of 'routed packaged component', it shall have exactly one shape_aspect that has exactly one property_definition that is associated to exactly one representation.

5.2.3.1.246 packaged_connector

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A packaged_connector is a packaged_part that implements the ARM concept of Packaged_connector (see 4.2.587).

EXPRESS specification:

```

*)
ENTITY packaged_connector
  SUBTYPE OF (packaged_part);
WHERE

  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP' IN
    TYPEOF (sa)))) = 1))) >= 1;

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (SIZEOF (QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'interface plane')) = 1)
    )) = 1)
    )
    ) <= 1;

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE' IN
    TYPEOF (sa)))) = 1))) <= 1;

END_ENTITY; -- packaged_connector
( *

```

Formal propositions:

WR1: The packaged_connector shall have at least one shape_aspect that is a packaged_connector-terminal_relationship.

WR2: The packaged_connector shall have at most one shape_aspect that is a seating plane in the role of 'interface plane'.

WR3: No more than one seating plane shall reference this packaged_connector through the of_shape attribute.

NOTE Normally the seating_plane that is related through the shape_aspect_relationship will also reference this packaged_connector.

5.2.3.1.247 packaged_connector_terminal_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A packaged_connector_terminal_relationship is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Packaged_connector_terminal_relationship (see 4.2.590).

EXPRESS specification:

```
*)
ENTITY packaged_connector_terminal_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE

  WR1: (SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'])
    = 1) AND
    (SELF.of_shape.definition\product_definition.description =
    'packaged connector');

  WR2: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART_TERMINAL' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect)) AND
    (SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description
    = 'interface terminal'));
```



```

WR3: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART_TERMINAL' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect)) AND
        (SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description
         = 'join terminal'));
END_ENTITY; -- packaged_connector_terminal_relationship
(*)

```

Formal propositions:

WR1: The packaged_connector_terminal_relationship shall be an aspect of the shape of a packaged_part, externally_defined_packaged_part, or library_defined_packaged_part with a description of 'packaged connector'.

WR2: The packaged_connector_terminal_relationship shall have a relating_shape_aspect that is a packaged_part_terminal with a description of 'interface terminal'.

WR3: The packaged_connector_terminal_relationship shall have a related_shape_aspect that is a packaged_part_terminal with a description of 'join terminal'.

5.2.3.1.248 packaged_part

A packaged_part is a physical_unit that implements the ARM concepts of Altered_packaged_part (see 4.2.10) and Packaged_part (see 4.2.591).

EXPRESS specification:

```

*)
ENTITY packaged_part
  SUPERTYPE OF (packaged_connector)
  SUBTYPE OF (physical_unit);
WHERE

WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT.OF_SHAPE') |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PACKAGED_PART_TERMINAL' IN
  TYPEOF (sa))) >= 0))) = 0;

WR2: SIZEOF (USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1;

```

```

WR3: SIZEOF (QUERY (ifu <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'implemented function') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT' IN
    TYPEOF (ifu.relatering_product_definition)) AND
    (ifu.relatering_product_definition.frame_of_reference.name =
    'functional design usage')))) <= 1;

WR4: NOT(EXISTS(SELF\product_definition.description)) OR
    ((NOT (SELF\product_definition.description = 'altered packaged part')) OR
    (SIZEOF (QUERY (bpp <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'base packaged part') |
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'] *
    TYPEOF (bpp.relatering_product_definition)) = 1) AND
    (bpp.relatering_product_definition.frame_of_reference.name =
    'physical design usage')))) >= 1));

WR5: SIZEOF (QUERY (upkg <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'used package') |
    SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGE'] *
    TYPEOF (upkg.relatering_product_definition)) = 1)) >= 1;

WR6: SELF.frame_of_reference.name = 'physical design usage';
END_ENTITY; -- packaged_part
(*

```

Formal propositions:

WR1: The packaged_part may have zero or more shape_aspect that is a packaged_part_terminal.

WR2: The packaged_part shall have no more than one material_designation.

WR3: The `packaged_part` shall be related to exactly one `functional_unit` with a `frame_of_reference` name of 'functional design usage' as the 'implemented function'.

WR4: If the `packaged_part` has a description of 'altered packaged part', it shall be related to at least one `packaged_part`, `externally_defined_packaged_part`, or `library_defined_packaged_part` with a `frame_of_reference` name of 'physical design usage' as the 'member connected terminal'.

WR5: The `packaged_part` shall be related to at least one `package`, `externally_defined_package`, or `library_defined_package` as the 'used package'.

NOTE The global rule `Packaged_part_unique_constraint` ensures that all packages so referenced are for the same `product_definition_formation`.

5.2.3.1.249 `packaged_part_terminal`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `packaged_part_terminal` is a `shape_aspect` that implements the ARM concepts of:

- `Packaged_part_interface_terminal` (see 4.2.592);
- `Packaged_part_join_terminal` (see 4.2.593);
- `Packaged_part_terminal` (see 4.2.594).

EXPRESS specification:

```

*)
ENTITY packaged_part_terminal
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\shape_aspect.description IN ['interface terminal',
    'join terminal'];

  WR2: (SIZEOF (TYPEOF (SELF.of_shape.definition) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'])
    = 1) AND
    (SELF.of_shape.definition\product_definition.frame_of_reference.name =
    'physical design usage');
```

```

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation))) >= 1))) = 0;

WR4: SIZEOF (QUERY (top <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'terminal of package') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_TERMINAL' IN
    TYPEOF (top.related_shape_aspect))) >= 1;

WR5: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'member connected terminal') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION'
    IN TYPEOF (mct.relying_shape_aspect))) <= 1;

WR6: (NOT(SELF\shape_aspect.description = 'interface terminal')) OR
    (SIZEOF(QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP' IN TYPEOF(sar))
    AND (sar.relying_shape_aspect\shape_aspect.description =
    'join terminal')))) >= 1);

WR7: (NOT(SELF\shape_aspect.description = 'join terminal')) OR
    (SIZEOF(QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP' IN TYPEOF(sar))
    AND (sar.relying_shape_aspect\shape_aspect.description =
    'interface terminal')))) <= 1);
END_ENTITY; -- packaged_part_terminal
(*

```

Formal propositions:

WR1: The packaged_part_terminal shall have a description of either 'interface terminal' or 'join terminal'.

WR2: The packaged_part_terminal shall be an aspect of the shape of a packaged_part, externally_defined_packaged_part, or library_defined_packaged_part that has a frame_of_reference name of 'physical design usage'.

WR3: The packaged_part_terminal shall have a property that is represented by at least one shape_representation.

WR4: The packaged_part_terminal shall relate to at least one package_terminal to specify the 'terminal of package'.

WR5: The packaged_part_terminal shall be related to at most one part_connected_terminals_definition as a member connected terminal.

WR6: If the packaged_part_terminal has a description of 'interface terminal', it shall be related to at least one packaged_part_terminal with a description of 'join terminal' by a packaged_connector_terminal_relationship.

WR7: If the packaged_part_terminal has a description of 'join terminal', it shall be related to at most one packaged_part_terminal with a description of 'interface terminal' by a packaged_connector_terminal_relationship.

NOTE Part_connected_terminals_definition provides a capability to state that multiple interface terminals are connected internally in the connector.

5.2.3.1.250 padstack_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A padstack_definition is a product_definition that implements the ARM concept Padstack_definition (see 4.2.595) and its subtypes.

EXPRESS specification:

```

*)
ENTITY padstack_definition
  SUPERTYPE OF (ONEOF( design_layer_type_specific_padstack_definition,
                        stratum_occurrence_specific_padstack_definition,
                        stratum_type_independent_padstack_definition))
  SUBTYPE OF (product_definition);
WHERE
```

```

WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
  TYPEOF (SELF))) OR
  (SIZEOF (QUERY (docs <*
  SELF\product_definition_with_associated_documents.
    documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename'))) <= 1);

WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
  adta.role\date_time_role.name = 'creation date')) +
  SIZEOF (QUERY (ada <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_DATE_ASSIGNMENT.ITEMS') |
  ada.role\date_role.name = 'creation date'))) = 1;

WR3: SIZEOF (USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

WR4: SIZEOF (QUERY (apoa <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  apoa.role\person_and_organization_role.name = 'creator')) +
  SIZEOF (QUERY (apoa <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
  apoa.role\organization_role.name = 'creator'))) >= 1;

WR5: SIZEOF (USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

WR6: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\product_definition ||
  SELF\padstack_definition))) = 0;

WR7: SELF.frame_of_reference.name IN ['layout design usage' ];

WR9: SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
  'PRODUCTS') |
  prpc\product_category.name = 'template model')) = 1;
END_ENTITY; -- padstack_definition
(*

```

Formal propositions:

WR1: If the padstack_definition is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR2: The padstack_definition shall have exactly one creation date.

WR3: The padstack_definition shall have exactly one approval.

WR4: The padstack_definition shall have at least one organization or person_and_organization as the creator.

WR5: The padstack_definition shall have exactly one security_classification.

WR6: The padstack_definition shall not be any other type of product_definition.

WR7: The padstack_definition shall have a frame_of_reference with a name of 'layout design usage'.

WR8: The padstack_definition shall be associated with a product that is a template model.

5.2.3.1.251 padstack_occurrence

A padstack_occurrence is an assembly_group_component_shape_aspect that implements the ARM concept of Padstack_occurrence (see 4.2.599).

EXPRESS specification:

```
*)
ENTITY padstack_occurrence
  SUBTYPE OF (assembly_group_component_shape_aspect);
END_ENTITY; -- padstack_occurrence
(*
```

5.2.3.1.252 padstack_occurrence_shape_aspect_relationship

A padstack_occurrence_shape_aspect_relationship is a shape_aspect_relationship that implements the ARM concept of Padstack_occurrence_sub_assembly_relationship (see 4.2.600)

EXPRESS specification:

```
*)
ENTITY padstack_occurrence_shape_aspect_relationship
  SUBTYPE OF (shape_aspect_relationship);
WHERE
```

```
  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relatng_shape_aspect;
```

```
WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
      SELF\shape_aspect_relationship ||
      SELF\padstack_occurrence_shape_aspect_relationship)) = 0;
```

```
WR3: SELF\shape_aspect_relationship.name = 'padstack occurrence sub assembly relationship'
END_ENTITY; -- padstack_occurrence_shape_aspect_relationship
(*
```

Formal propositions:

WR1: The related shape_aspect shall not be the relating shape_aspect.

WR2: The types of padstack_occurrence_shape_aspect_relationship shall be only padstack_occurrence_shape_aspect_relationship, shape_aspect, and shape_aspect_relationship.

WR3: The name attribute of the shape_aspect_relationship supertype shall be 'padstack occurrence sub assembly relationship'.

5.2.3.1.253 parallelism_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A parallelism_tolerance is a geometric_tolerance_with_specified_datum_source that implements the ARM concept of Parallelism_tolerance (see 4.2.602).

EXPRESS specification:

```
*)
ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE
```

```
WR1: SELF\geometric_tolerance.name = 'parallelism';
END_ENTITY; -- parallelism_tolerance
(*
```

Formal propositions:

WR1: The name for the parallelism_tolerance shall be 'parallelism'.

Informal propositions:

IP1: The `toleranced_shape_aspect` of the `parallelism_tolerance` shall either lie on the boundary of the shape of the product; be a `centre_of_symmetry` with a description of either 'axis' or 'plane'; be a tangent with a description of 'plane'; be a `part_template_definition`; or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which either are all of type `part_template_definition`, are all of type `centre_of_symmetry` and have the same description – either 'axis' or 'plane', or are all of type tangent with a description of 'plane'.

IP2: The `parallelism_tolerance` shall have at most two `physical_unit_datum` in its referenced `datum_system`.

IP3: The `parallelism_tolerance` shall have at least one `physical_unit_datum` in its referenced `datum_system` that has description of either 'axis' or 'plane'.

IP4: If the `toleranced_shape_aspect` of the `parallelism_tolerance` is neither a `centre_of_symmetry` with a description of 'axis' nor a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'axis', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

5.2.3.1.254 parameter_assignment_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `parameter_assignment_representation` is a representation that implements the ARM concept of `Parameter_assignment` (see 4.2.603).

EXPRESS specification:

```

*)
ENTITY parameter_assignment_representation
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (SELF.items) = 2;

  WR2: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (TYPEOF (it) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODEL_PARAMETER'])
    = 1))) = 1;

  WR3: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (TYPEOF (it) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COORDINATED_REPRESENTATION_ITEM'])
    = 1))) <= 1;

```

```

WR4: sizeof (QUERY (it <* SELF.items |
    (sizeof (QUERY(aga <* USEDIN (it,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
            (sizeof (typeof ( aga\group_assignment.assigned_group) *
                ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CHARACTERISTIC_TYPE']) = 1)))
    = 1))) <= 1;

WR5: sizeof(typeof(SELF) - typeof(SELF\representation ||
    SELF\parameter_assignment_representation)) = 0;
END_ENTITY; -- parameter_assignment_representation
(*

```

Formal propositions:

WR1: The parameter_assignment_representation shall have exactly two representation_items.

WR2: The parameter_assignment_representation shall have exactly one item that is a model_parameter.

WR3: The parameter_assignment_representation shall have no more than one item that is a coordinated_characteristic.

WR4: The parameter_assignment_representation shall have no more than one item that is a characteristic.

WR5: The parameter_assignment_representation shall not be any other type other than a representation.

5.2.3.1.255 parameter_type

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A parameter_type is a group that specifies a boolean_property_type, a logical_property_type, a physical_property_type or a string_property_type for a model_parameter.

EXPRESS specification:

```

*)
ENTITY parameter_type
    SUBTYPE OF (group);
WHERE

    WR1: SELF\group.name IN ['string property type', 'logical property type',
        'physical property type', 'boolean property type'];
END_ENTITY; -- parameter_type
(*

```

Formal propositions:

WR1: A parameter_type shall have a name that is 'string property type', 'logical property type', 'physical property type', or 'boolean property type'.

5.2.3.1.256 part_connected_terminals_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A part_connected_terminals_definition is a shape_aspect that implements the ARM concept of Part_connected_terminals_definition (see 4.2.605).

EXPRESS specification:

```

*)
ENTITY part_connected_terminals_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1: SELF\shape_aspect.name;
  WHERE

    WR1: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'member connected terminal') |
    SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_MODULE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTERCONNECT_MODULE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART_TERMINAL'] *
      TYPEOF (mct.related_shape_aspect)) = 1)) >= 2;
END_ENTITY; -- part_connected_terminals_definition
(

```

Formal propositions:

UR1: The name shall be unique.

WR1: The part_connected_terminals_definition shall relate to at least two instances of assembly_module_terminal, bare_die_terminal, interconnect_module_terminal, package_terminal, or packaged_part_terminal to specify a 'member connected terminal'.

Informal propositions:

IP1: Each terminal related by this part_connected_terminals_definition shall be a terminal of the same product version.

5.2.3.1.257 part_interface_access_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A part_interface_access_feature is a shape_aspect that implements the ARM concept of Part_interface_access_feature (see 4.2.614).

EXPRESS specification:

```
*)
ENTITY part_interface_access_feature
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'interface_access area') AND
    (sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) = 1;
END_ENTITY; -- part_interface_access_feature
(*
```

Formal propositions:

WR1: The part_interface_access_feature shall be related as the 'interface_access area' to exactly one shape_aspect with a description of 'connection zone'.

5.2.3.1.258 part_mating_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A part_mating_feature is a shape_aspect that implements the ARM concept of Part_mating_feature (see 4.2.617).

EXPRESS specification:

```

*)
ENTITY part_mating_feature
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'mating area') AND
    (sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) = 1;
END_ENTITY; -- part_mating_feature
( *

```

Formal propositions:

WR1: The part_mating_feature shall be related as the 'mating area' to exactly one shape_aspect with a description of 'connection zone'.

5.2.3.1.259 part_mounting_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A part_mounting_feature is a shape_aspect that implements the ARM concept of Part_mounting_feature (see 4.2.618).

EXPRESS specification:

```

*)
ENTITY part_mounting_feature
  SUPERTYPE OF (bare_die_surface)
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'mounting area') AND
    (sar.related_shape_aspect\shape_aspect.description = 'connection zone')))) = 1;
END_ENTITY; -- part_mounting_feature
( *

```

Formal propositions:

WR1: The part_mounting_feature shall be related as the 'mounting area' to exactly one shape_aspect with a description of 'connection zone'.

5.2.3.1.260 part_template_definition

A part_template_definition is a shape_aspect that implements the ARM concepts of:

- Blind_passage_template (see 4.2.70);
- Component_termination_passage_template (see 4.2.150);
- Counterbore_passage_template (see 4.2.198);
- Countersunk_passage_template (see 4.2.199);
- Default_trace_template (see 4.2.252);
- Dependent_material_removal_feature_template (see 4.2.258);
- Inter_stratum_feature_template (see 4.2.435);
- Material_addition_feature_template (see 4.2.529);
- Material_removal_feature_template (see 4.2.534);
- Non_conductive_cross_section_template (see 4.2.558);
- Part_template (see 4.2.620);
- Part_text_template (see 4.2.632);
- Physical_feature_or_part_template (see 4.2.658);
- Printed_connector_template (see 4.2.726);
- Printed_part_cross_section_template (see 4.2.728);
- Printed_part_template (see 4.2.730);
- Special_symbol_part_template (see 4.2.835);
- Stratum_feature_template (see 4.2.861);
- Trace_template (see 4.2.930);

- Unsupported_passage_template (see 4.2.941);
- Via_template (see 4.2.952).

EXPRESS specification:

```

*)
ENTITY part_template_definition
  SUPERTYPE OF (dependent_material_removal_feature_template)
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\
    shape_aspect.of_shape\
    property_definition.definition.frame_of_reference\
    application_context_element.name = 'template definition';

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANALYTICAL_MODEL' IN
    TYPEOF (pdr.used_representation)) AND
    (pdr.used_representation\representation.name = 'part template analytical model'))))
    <= 1))) = 0;

  WR3: (NOT (SELF\shape_aspect.description =
    'component termination passage template')) OR
    (SIZEOF (QUERY (ctpt <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'inter stratum feature passage technology') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PASSAGE_TECHNOLOGY' IN
    TYPEOF (ctpt.relateing_shape_aspect)) AND
    (ctpt.relateing_shape_aspect\shape_aspect.description =
    'default component termination passage definition')))) = 1));

```

```

WR4: (NOT (SELF\shape_aspect.description =
    'component termination passage template')) OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'access mechanism') |
    am.related_shape_aspect\shape_aspect.description IN
    ['component termination passage template interface terminal',
    'component termination passage template join terminal']))) >= 2);

WR5: (NOT (SELF\shape_aspect.description = 'default trace template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'technology usage') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_TECHNOLOGY' IN
    TYPEOF (tu.relating_property_definition.definition))) = 1)) = 1);

WR6: (NOT (SELF\shape_aspect.description = 'inter stratum feature template')) OR
(SIZEOF (QUERY (isfpt <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'inter stratum feature passage technology') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PASSAGE_TECHNOLOGY' IN
    TYPEOF (isfpt.relating_shape_aspect)))) = 1);

WR7: (NOT (SELF\shape_aspect.description = 'printed connector template')) OR
(SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name IN
    ['connector']))) >= 1);

```



```

WR8: (NOT (SELF\shape_aspect.description = 'printed part cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ANALYTICAL_MODEL' IN
TYPEOF (pdr.used_representation)) AND
(pdr.used_representation\representation.name = 'transmission line model'))
= 1))) = 0));

```

```

WR9: (NOT (SELF\shape_aspect.description = 'printed part cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2)) = 2)) = 1))) = 0));

```

```

WR10: (NOT (SELF\shape_aspect.description = 'printed part cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name = 'maximum width'))
= 1
)) = 1))) = 0));

```

```

WR11: (NOT (SELF\shape_aspect.description = 'printed part cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name = 'minimum width')))) = 1
)) = 1))) = 0);

```

```

WR12: (NOT (SELF\shape_aspect.description = 'printed part template')) OR
(SIZEOF (QUERY (impl_func <* QUERY (pdr <*
USEDIN (SELF.of_shape.definition,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') |
pdr\product_definition_relationship.name = 'implemented function') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT' IN
TYPEOF (impl_func.relying_product_definition)) AND
(impl_func.relying_product_definition.frame_of_reference.name =
'functional design usage')))) = 1);

```

```

WR13: (NOT (SELF\shape_aspect.description = 'printed part template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELYING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'associated definition') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL' IN
TYPEOF (ad.related_shape_aspect)) AND
(ad.related_shape_aspect\shape_aspect.description IN
['interface terminal', 'join terminal']))) >= 2);

```

```

WR14: (NOT (SELF\shape_aspect.description = 'trace template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
(SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name = 'curve style parameters'
)) = 1 ) )) = 1 );

WR15: (NOT (SELF\shape_aspect.description = 'unsupported passage template')) OR
(SIZEOF (QUERY (upt <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'inter stratum feature passage technology') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN
TYPEOF (upt.relate_shape_aspect)) AND
(upt.relate_shape_aspect\shape_aspect.description =
'default unsupported passage definition')))) = 1);

WR16: (NOT (SELF\shape_aspect.description = 'via template')) OR
(SIZEOF (QUERY (vpt <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'inter stratum feature passage technology') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN
TYPEOF (vpt.relate_shape_aspect)) AND
(vpt.relate_shape_aspect\shape_aspect.description =
'default via definition')))) = 1);

WR17: (NOT (SELF\shape_aspect.description = 'via template')) OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'access mechanism') |
am.related_shape_aspect\shape_aspect.description =
'via template terminal')) >= 2);

WR18: EXISTS(SELF\shape_aspect.name);

```

```

WR19: (NOT (SELF\shape_aspect.description = 'non conductive cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'technology usage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY' IN
TYPEOF (tu.relatng_property_definition.definition)))) = 1)) = 1);

```

```

WR20: (NOT (SELF\shape_aspect.description = 'non conductive cross section template')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it\representation_item.name = 'nominal width')))) = 1
)) = 1))) = 0);

```

```

WR21: (NOT (SELF\shape_aspect.description = 'printed connector template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATNG_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'associated definition') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL' IN
TYPEOF (ad.related_shape_aspect)) AND
(ad.relatng_shape_aspect\shape_aspect.description IN
['interface terminal', 'join terminal']))) >= 2);

```

```

WR22: (NOT (SELF\shape_aspect.description =
            'printed part cross section template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'horizontal material link') |
NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (ad.relatng_shape_aspect)) AND
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
TYPEOF (ad.relatng_shape_aspect\shape_aspect_relationship.
related_shape_aspect)) AND
(ad.relatng_shape_aspect\shape_aspect_relationship.
related_shape_aspect\shape_aspect.description = 'left'))))) = 0);

```

```

WR27: (NOT (SELF\shape_aspect.description =
            'printed part cross section template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'horizontal material link') |
NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (ad.relatng_shape_aspect)) AND
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
TYPEOF (ad.relatng_shape_aspect\shape_aspect_relationship.
relating_shape_aspect)) AND
(ad.relatng_shape_aspect\shape_aspect_relationship.
relating_shape_aspect\shape_aspect.description = 'right'))))) = 0);

```

```

WR28: (NOT (SELF\shape_aspect.description =
            'printed part cross section template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'vertical material link') |
NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (ad.relatng_shape_aspect)) AND
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
TYPEOF (ad.relatng_shape_aspect\shape_aspect_relationship.
related_shape_aspect)) AND
(ad.relatng_shape_aspect\shape_aspect_relationship.
related_shape_aspect\shape_aspect.description = 'bottom'))))) = 0);

```

```

WR29: (NOT (SELF\shape_aspect.description =
            'printed part cross section template')) OR
(SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'vertical material link') |
NOT(( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (ad.relatng_shape_aspect)) AND
(( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
TYPEOF (ad.relatng_shape_aspect\shape_aspect_relationship.
relatng_shape_aspect)) AND
(ad.relatng_shape_aspect\shape_aspect_relationship.
relatng_shape_aspect\shape_aspect.description = 'top'))))) = 0);

WR30: SELF\shape_aspect.name = 'NULL';
END_ENTITY; -- part_template_definition
(*)

```

Formal propositions:

WR1: The part_template_definition shall be an aspect of the shape of a product_definition that has an application_context_element with a name of 'template definition'.

WR2: The part_template_definition shall have at most one representation that is an analytical_model and has a name of 'part template analytical model'.

WR3: If the part_template_definition has a description of 'component termination passage template', it shall be related to a passage_technology with a description of 'default component termination passage definition' as a 'inter stratum feature passage technology'.

WR4: If the part_template_definition has a description of 'component termination passage template', it shall be used to relate at least two shape_aspects with a description of either 'component termination passage template interface terminal' or 'component termination passage template join terminal' to specify the 'access mechanism'.

WR5: If the part_template_definition has a description of 'default trace template' it shall have a property that is related to a property of a stratum_technology as the 'technology usage'.

WR6: If the part_template_definition has a description of 'inter stratum feature template', it shall be related to a passage_technology as an 'inter stratum feature passage technology'.

WR7: If the part_template_definition has a description of 'printed connector template', it shall be used to relate at least one shape_aspect to specify a 'connector'.

WR8: If the part_template_definition has a description of 'printed part cross section template', it shall be represented by exactly one analytical_model with a name of 'transmission line model'.

WR9: If the part_template_definition has a description of 'printed part cross section template', it shall have a representation that contains exactly two length_measure_with_units.

WR10: If the `part_template_definition` has a description of 'printed part cross section template', it shall have a representation that contains exactly one `length_measure_with_unit` with a name of 'maximum width'.

WR11: If the `part_template_definition` has a description of 'printed part cross section template', it shall have a representation that contains exactly one `length_measure_with_unit` with a name of 'minimum width'.

WR12: If the `part_template_definition` has a description of 'printed part template', it shall be an aspect of the shape of a `product_definition` that is the 'implemented function' for exactly one `functional_unit` with a `frame_of_reference` name of 'functional design usage'.

WR13: If the `part_template_definition` has a description of 'printed part template', it shall be used to relate two or more `printed_part_template_terminals` with a description of either 'interface terminal' or 'join terminal' to specify an 'associated definition'.

WR14: If the `part_template_definition` has a description of 'trace template', it shall have a representation that contains exactly one `styled_item` with a name of 'trace style' that specifies a `curve_style`.

WR15: If the `part_template_definition` has a description of 'unsupported passage template', it shall be related to exactly one `passage_technology` with a description of 'default unsupported passage definition'.

WR16: If the `part_template_definition` has a description of 'via template', it shall be related to exactly one `passage_technology` with a description of 'default via definition'.

WR17: If the `part_template_definition` has a description of 'via template', it shall relate at least two `shape_aspects` with a description of 'via template terminal' as the 'access mechanism'.

WR18: The `part_template_definition` name shall exist.

NOTE This is a placeholder rule.

WR19: If the `part_template_definition` has a description of 'non conductive cross section template' it shall have a property that is related to a property of a `stratum_technology` as the 'technology usage'.

WR20: If the `part_template_definition` has a description of 'non conductive cross section template', it shall have a representation that contains exactly one `length_measure_with_unit` with a name of 'nominal width'.

WR21: If the `part_template_definition` has a description of 'printed connector template', it shall be used to relate two or more `printed_part_template_terminals` with a description of either 'interface terminal' or 'join terminal' to specify an 'associated definition'.

WR22: If the `part_template_definition` has a description of 'printed part cross section template', and if there are horizontally adjacent disjoint materials, the subsequent boundary shall be the 'right' boundary for each material link between two materials.

WR23: If the `part_template_definition` has a description of 'printed part cross section template', and if there are horizontally adjacent disjoint materials, the precedent boundary shall be the 'left' boundary for each material link between two materials.

WR24: If the `part_template_definition` has a description of 'printed part cross section template', and if there are vertically adjacent disjoint materials, the subsequent boundary shall be the 'top' boundary for each material link between two materials.

WR25: If the `part_template_definition` has a description of 'printed part cross section template', and if there are vertically adjacent disjoint materials, the precedent boundary shall be the 'bottom' boundary for each material link between two materials.

WR26: The name inherited from `shape_aspect` shall be 'NULL'.

5.2.3.1.261 `part_text_template_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `part_text_template_definition` is a `part_template_definition` that implements the ARM concept of `Part_text_template` (see 4.2.632) .

EXPRESS specification:

*)

```
ENTITY part_text_template_definition
  SUBTYPE OF (part_template_definition);
WHERE
```

```
WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'TEXT_LITERAL' IN
  TYPEOF (it))) = 1)) = 1))) = 0);
```

```
WR2: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  (SIZEOF (
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2) AND (it\representation_item.name =
  'maximum font vertical extent')) = 1
  )) = 1))) = 0);
```



```

WR3: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  (SIZEOF (
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2) AND (it\representation_item.name =
    'maximum font horizontal extent')) = 1
  )) = 1))) = 0);
END_ENTITY; -- part_text_template_definition
(*

```

Formal propositions:

WR1: The `part_text_template_definition` shall have a representation that has exactly one `text_literal`.

WR2: The `part_text_template_definition` shall have a representation that has exactly one `measure_representation_item` that is a `length_measure_with_unit` with a name of 'maximum font vertical extent'.

WR3: The `part_text_template_definition` shall have a representation that has exactly one `measure_representation_item` that is a `length_measure_with_unit` with a name of 'maximum font horizontal extent'.

5.2.3.1.262 **part_tooling_feature**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `part_tooling_feature` is a `shape_aspect` that implements the ARM concept of `Part_tooling_feature` (see 4.2.633).

EXPRESS specification:

```

*)
ENTITY part_tooling_feature
  SUPERTYPE OF (fiducial_part_feature ANDOR
                test_point_part_feature)
  SUBTYPE OF (shape_aspect);
END_ENTITY; -- part_tooling_feature
(*

```

5.2.3.1.263 **passage_technology**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `passage_technology` is a `shape_aspect` that implements the ARM concepts of:

- `Default_component_termination_passage_definition` (see 4.2.249);
- `Default_unsupported_passage_definition` (see 4.2.254);
- `Passage_technology` (see 4.2.638).

EXPRESS specification:

```

*)
ENTITY passage_technology
  SUBTYPE OF (shape_aspect);
UNIQUE
  UR1: SELF\shape_aspect.name, SELF\shape_aspect.of_shape;
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);

  WR2: SIZEOF (USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1;

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation\representation.name =
        'physical characteristics representation')) = 1))) = 0;

  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        pdr.used_representation\representation.name =
        'physical characteristics representation') |
        NOT (
          {1 <= SIZEOF (pcr.used_representation.items) <= 8}
          ))) = 0))) = 0;

```

```

WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
    NOT (it\representation_item.name IN ['maximum aspect ratio', 'plated passage',
    'maximum allowed component terminal extent',
    'minimum allowed component terminal extent',
    'maximum as finished deposition thickness',
    'minimum as finished deposition thickness',
    'maximum as finished passage extent',
    'minimum as finished passage extent'])))) = 0))) = 0))) = 0;

WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
    (SIZEOF (
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RATIO_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it\representation_item.name = 'maximum aspect ratio') AND
    (it\measure_with_unit.value_component > 1.0))) <= 1))) = 0))) = 0;

WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND
    (it\representation_item.name = 'plated passage') AND
    (it\descriptive_representation_item.description IN ['true', 'false']))) = 1))) = 0)

```

```

WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      (SIZEOF (
        ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it\representation_item.name = 'minimum as finished passage extent')
      )) <= 1))) = 0))) = 0;

```

```

WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      (SIZEOF (
        ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it\representation_item.name =
        'maximum as finished deposition thickness')) <= 1))) = 0))) = 0;

```

```

WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
' physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
[ ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MEASURE_REPRESENTATION_ITEM',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
' minimum as finished deposition thickness')))) <= 1))) = 0))) = 0;

WR11: (NOT (SELF\shape_aspect.description IN
[ ' default component termination passage definition',
' default via definition'])) OR
(SIZEOF (USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MATERIAL_DESIGNATION.DEFINITIONS')) = 1);

WR12: (NOT (SELF\shape_aspect.description IN
[ ' default component termination passage definition',
' default unsupported passage definition',
' default via definition'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
' physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
[ ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MEASURE_REPRESENTATION_ITEM',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name = ' minimum as finished passage extent'))))
= 1))) = 0))) = 0);

```

```

WR13: (NOT (SELF\shape_aspect.description =
'default component termination passage definition')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
'minimum as finished deposition thickness'))
= 1))) = 0))) = 0);

WR14: (NOT (SELF\shape_aspect.description =
'default component termination passage definition')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
'minimum allowed component terminal extent')) = 1))) = 0))) = 0);

```

```

WR15: (NOT (SELF\shape_aspect.description =
'default component termination passage definition')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
'maximum allowed component terminal extent')))) = 1))) = 0))) = 0);

WR16: (NOT (SELF\shape_aspect.description = 'default via definition')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name =
'minimum as finished deposition thickness')))) = 1))) = 0))) = 0);

```

```

WR17: sizeof (QUERY (rpt <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'as finished inter stratum extent') |
    (rpt.related_shape_aspect.of_shape\property_definition.description =
    'finished stratum extent') AND
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP'
    IN
    TYPEOF (rpt.related_shape_aspect.of_shape\property_definition.
    definition))
    AND
    (rpt.related_shape_aspect.of_shape\property_definition.
    definition\product_definition_relationship.name =
    'inter stratum extent')))) = 1;
END_ENTITY; -- passage_technology
(*

```

Formal propositions:

UR1: The name shall be unique within the scope of the referenced shape.

WR1: The passage_technology shall be an aspect of the shape of a product_definition.

WR2: The passage_technology shall have at most one material designation.

WR3: The passage_technology shall have exactly one representation with a name of 'physical characteristics representation'.

WR4: The passage_technology shall have at least one and at most eight items in its representation with a name of 'physical characteristics representation'.

WR5: The items in the representation with a name of 'physical characteristics representation' of a passage_technology shall have a name of either 'maximum aspect ratio', 'plated passage', 'maximum allowed component terminal extent', 'minimum allowed component terminal extent', 'maximum as finished deposition thickness', 'minimum as finished deposition thickness', 'maximum as finished passage extent', or 'minimum as finished passage extent'.

WR6: At most one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a ratio_measure_with_unit with a name of 'maximum aspect ratio' and a value greater than 1.0.

WR7: Exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a descriptive_representation_item with a name of 'plated passage' and a description of 'true' or 'false'.

WR8: At most one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum as finished passage extent'.

WR9: At most one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'maximum as finished deposition thickness'.

WR10: At most one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum as finished deposition thickness'.

WR11: If the passage_technology has a description of 'default via definition' or 'default component termination passage definition', it shall have exactly one material_designation.

WR12: If the passage_technology has a description of 'default component termination passage definition', 'default unsupported passage definition' or 'default via definition', exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum as finished passage extent'.

WR13: If the passage_technology has a description of 'default component termination passage definition', exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum as finished deposition thickness'.

WR14: If the passage_technology has a description of 'default component termination passage definition', exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum allowed component terminal extent'.

WR15: If the passage_technology has a description of 'default component termination passage definition', exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'maximum allowed component terminal extent'.

WR16: If the passage_technology has a description of 'default via definition', exactly one item in the representation with a name of 'physical characteristics representation' of a passage_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum as finished deposition thickness'.

WR17: The passage_technology shall relate to exactly one shape_aspect of a shape with a description of 'finished stratum extent' of a product_definition_relationship with a name of 'inter stratum extent' to specify the 'as finished inter stratum extent'.

Informal propositions:

IP1: If the passage_technology is related to a inter_stratum_feature with a description of 'interconnect module edge' or 'interconnect module edge segment' it, shall not be related to a representation_item with a name of 'maximum as finished passage extent'

IP2: If the item in the representation with a name of 'physical characteristics representation' of a passage_technology has a descriptive_representation_item with a name of 'plated passage' and a description of 'true' then the material designation shall have be the designation for a material_designation_characterization with a property that is a material_property_representation that has a definition that is a material_property and a used_representation with a name of 'conductivity classification representation' that has an item that is a descriptive_representation_item with a name of 'electrical conductivity classification' and a description of 'conductive'.

5.2.3.1.264 perpendicularity_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A perpendicularity_tolerance is a geometric_tolerance_with_specified_datum_source that implements the ARM concept of Perpendicularity_tolerance (see 4.2.645).

EXPRESS specification:

```
*)
ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'perpendicularity';
END_ENTITY; -- perpendicularity_tolerance
(*
```

Formal propositions:

WR1: The name for the perpendicularity_tolerance shall be 'perpendicularity'.

Informal propositions:

IP1: The toleranced_shape_aspect of the perpendicularity_tolerance shall either lie on the boundary of the shape of the product; be a centre_of_symmetry with a description of either 'axis' or 'plane'; be a tangent with a description of 'plane'; be a part_template_definition; or be a composite_shape_aspect the 'composing' shape_aspects of which either are all of type part_template_definition, are all of type centre_of_symmetry and have the same description – either 'axis' or 'plane', or are all of type tangent with a description of 'plane'.

IP2: The perpendicularity_tolerance shall have at most two physical_unit_datum in its referenced datum_system.

IP3: The perpendicularity_tolerance shall have at least one physical_unit_datum in its referenced datum_system that has description of either 'axis' or 'plane'.

IP4: If the `toleranced_shape_aspect` of the `perpendicularity_tolerance` is neither a `centre_of_symmetry` with a description of 'axis' nor a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'axis', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

5.2.3.1.265 `physical_connectivity_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_connectivity_definition` is a `shape_aspect` that implements the ARM concepts of `Physical_connectivity_definition` (see 4.2.652) and `Physical_connectivity_structure_definition` (see 4.2.657).

EXPRESS specification:

```
*)
ENTITY physical_connectivity_definition
  SUBTYPE OF (shape_aspect);
UNIQUE
  UR1: SELF\shape_aspect.name, SELF\shape_aspect.of_shape;
WHERE

  WR1: SIZEOF (QUERY (at <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated terminals') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (at.related_shape_aspect)) AND
    ((at.related_shape_aspect\shape_aspect.description =
      'assembly module component terminal') OR
    (at.related_shape_aspect\shape_aspect.description =
      'bare die component terminal') OR
    (at.related_shape_aspect\shape_aspect.description =
      'interconnect component join terminal') OR
    (at.related_shape_aspect\shape_aspect.description =
      'interconnect module component terminal') OR
    (at.related_shape_aspect\shape_aspect.description =
      'minimally defined component terminal') OR
    (at.related_shape_aspect\shape_aspect.description =
      'packaged component join terminal'))
    )) > 1;
```

```

WR2: SIZEOF (QUERY( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'connectivity allocation')) = 1)
    ))
    <= 1;
END_ENTITY; -- physical_connectivity_definition
(*

```

Formal propositions:

UR1: The name shall be unique within the context of the product_definition_shape in the role of of_shape for the shape_aspect supertype.

NOTE Since there may be only one product_definition for a product_definition_shape, this guarantees uniqueness within the context of the product_definition.

WR1: The physical_connectivity_definition shall relate to two or more physical_component_terminal to specify the 'associated terminals'.

WR2: The physical_connectivity_definition shall be related by at most one property_definition_relationship as a 'connectivity allocation'.

Informal propositions:

IP1: All component_terminals related to a physical_connectivity_definition shall be in the same design view.

IP2: All component_terminals related to a physical_connectivity_definition shall not be in the set of terminals of the components that implement the connectivity defined by the physical_connectivity_definition.

NOTE The terminals identified by the physical_connectivity_definition are a subset of the terminals referenced by the members of assembly_joint or by the members of the implementation of the ARM concept of Interface_mounted_join.

5.2.3.1.266 physical_connectivity_element

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A physical_connectivity_element is a shape_aspect_relationship and a shape aspect that implement the ARM concept of Physical_connectivity_element (see 4.2.654).

EXPRESS specification:

```

*)
ENTITY physical_connectivity_element
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
WHERE

  WR1: (SELF.relatng_shape_aspect\shape_aspect.description =
        'topological junction') XOR
        (SIZEOF ([
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'COMPONENT_TERMINAL'] *
          TYPEOF (SELF.relatng_shape_aspect)) = 1);

  WR2: (SELF.related_shape_aspect\shape_aspect.description = 'topological junction') XOR
        (SIZEOF ([
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'COMPONENT_TERMINAL'] *
          TYPEOF (SELF.related_shape_aspect)) = 1);

  WR3: SIZEOF (QUERY (se <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'structure element') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION' IN
    TYPEOF (se.relatng_shape_aspect))) = 1;

  WR4: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT' IN
    TYPEOF (SELF.of_shape.definition)) AND
    (SELF.of_shape.definition\product_definition.
      frame_of_reference.name =
      'physical design');
END_ENTITY; -- physical_connectivity_element
(

```

Formal propositions:

WR1: The `physical_connectivity_element` shall have a `relatng_shape_aspect` that either has a description of 'topological junction', or is a `component_terminal`.

WR2: The `physical_connectivity_element` shall have a `related_shape_aspect` that either has a description of 'topological junction', or is a `component_terminal`.

WR3: The `physical_connectivity_element` shall be related to exactly one `physical_connectivity_definition` as a 'structure element'.

WR4: The `physical_connectivity_element` shall be an aspect of the shape of a `physical_unit` that has a `frame_of_reference` name of 'physical design'.

5.2.3.1.267 `physical_network`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_network` is a `shape_aspect` that implements the ARM concept of `Physical_network` (see 4.2.660).

NOTE A `physical_network` should not be confused with the like sounding term `physical_unit_network_definition`.

EXPRESS specification:

```
*)
ENTITY physical_network
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (cr <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'connectivity requirement') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (cr.related_shape_aspect))) AND
    ((cr.related_shape_aspect\shape_aspect.description =
    'component termination passage join terminal') OR
    (cr.related_shape_aspect\shape_aspect.description =
    'land join terminal') OR
    (cr.related_shape_aspect\shape_aspect.description =
    'non functional land join terminal') OR
    (cr.related_shape_aspect\shape_aspect.description =
    'printed component join terminal'))
  )) >= 2;

  WR2: SIZEOF (QUERY (nt <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'network topology') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT' IN
    TYPEOF (nt.related_shape_aspect)))) >= 1;
```

```

WR3: sizeof (QUERY (nt <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'reference connected terminals') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION' IN
    TYPEOF (nt.relatng_shape_aspect))) <= 1;
END_ENTITY; -- physical_network
(*

```

Formal propositions:

WR1: The `physical_network` shall relate to at least two `component_terminals` with the specified descriptions to specify a 'connectivity requirement'.

WR2: The `physical_network` shall relate to at least one `join_shape_aspect` to specify a 'network topology'.

WR3: The `physical_network` shall relate to no more than one `part_connected_terminals_definition` to specify 'reference connected terminals'.

Informal propositions:

IP1: All `component_terminals` related to a `physical_network` in the role of `connectivity_requirement_element` shall be in the same design view as the `physical_network` (`interconnect_module`.)

IP2: All `join_relationships` related to a `physical_network` in the role of `interconnect_module_net_topology_element` shall be in the same design view as the `physical_network` (`interconnect_module`.)

IP3: All `interconnect_module_net_topology_element` and `connectivity_requirement_element` related to `physical_network` shall be in the same design view (`interconnect_module`.)

IP4: The `material_electrical_conductivity_category` shall be the same for all materials that participate in the `physical_network`.

IP5: The `physical_network` shall not be referenced by more than one `property_definition_relationship` that relates the `physical_network` to an aggregate connectivity requirement implementation.

5.2.3.1.268 `physical_node_requirement_to_implementing_component_allocation`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_node_requirement_to_implementing_component_allocation` is a `shape_aspect_relationship` and a `shape_aspect` that is the allocation of a physical node to the component that is partially or wholly responsible for implementing that node in the assembly. The relating and related `shape_aspects` are the node and a `shape_aspect` of the component respectively. A `physical_node_requirement_to_implementing_component_allocation` implements the ARM concept of `Physical_node_requirement_to_implementing_component_allocation` (see 4.2.663).

EXPRESS specification:

```

*)
ENTITY physical_node_requirement_to_implementing_component_allocation
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
WHERE

  WR1: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION' IN
    TYPEOF (SELF.related_shape_aspect.of_shape\property_definition.definition)) AND
    (NOT (SELF.related_shape_aspect.of_shape\property_definition.definition\product_def.
    ['laminate component'])));

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION' IN
    TYPEOF (SELF.relate_shape_aspect);

  WR3: acyclic_shape_aspect_relationship(SELF,
    [SELF\shape_aspect_relationship.related_shape_aspect],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_NODE_REQUIREMENT_TO_IMPLEMENTING_COMPONENT_ALLOCATION');

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\shape_aspect_relationship ||
    SELF\physical_node_requirement_to_implementing_component_allocation)) = 0;
(*)

  WR5: SIZEOF(terminal_of_implementing_component(SELF, SELF.relate_shape_aspect)) = 0;
*)
END_ENTITY; -- physical_node_requirement_to_implementing_component_allocation
(*)

```

Formal propositions:

WR1: The physical_node_requirement_to_implementing_component_allocation shall have a related_shape_aspect that references a component_definition that is not a laminate component.

WR2: The physical_node_requirement_to_implementing_component_allocation shall have a relating_shape_aspect that references a physical_connectivity_definition.

WR3: The members of physical_node_requirement_to_implementing_component_allocation shall form a directed acyclic graph with the relating_shape_aspect, related_shape_aspect attributes of physical_node_requirement_to_implementing_component_allocation.

WR4: A member of physical_node_requirement_to_implementing_component_allocation shall not be any other subtype of shape_aspect or shape_aspect_relationship.

WR5: in arm terms: SIZEOF(QUERY (pct <* physical_node_requirement.associated_terminals | pct\component_feature.associated_component = implementation_component)) = 0;

5.2.3.1.269 physical_unit

A physical_unit is a product_definition that implements the ARM concepts of:

- Assembly_module (see 4.2.43);
- Assembly_module_usage_view (see 4.2.51);
- Externally_defined_ee_product_definition (see 4.2.364);
- Ee_product_definition (see 4.2.323);
- Interconnect_module (see 4.2.440);
- Interconnect_module_usage_view (see 4.2.465);
- Library_defined_ee_product_definition (see ??);
- Physical_unit (see 4.2.664);
- Physical_unit_design_view (see 4.2.667);
- Physical_unit_usage_view (see 4.2.672).

EXPRESS specification:

```

*)
ENTITY physical_unit
  SUPERTYPE OF (ONEOF (bare_die,
                        package,
                        packaged_part,
                        assembly_definition,
                        interconnect_definition,
                        reference_packaged_part_assembly_implementation,
                        reference_packaged_part_interconnect_implementation))
  SUBTYPE OF (product_definition);
WHERE

  WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
            TYPEOF (SELF))) OR
        (SIZEOF (QUERY (docs <*
            SELF\product_definition_with_associated_documents.
            documentation_ids |
            docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);

```

```

WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) +
    SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ada.role\date_role.name = 'creation date')) = 1;

WR3: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

WR4: SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

WR5: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

WR6: SELF.frame_of_reference.name IN ['physical design',
    'physical design usage'];

WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (dut <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'physical unit technology') |
    dut.relatng_property_definition\property_definition.name =
    'unit technology')) = 1)) <= 1;

```

WR8: NOT(is_assembly_module_usage(SELF))
OR

```
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_TERMINAL' IN
TYPEOF (sa))) >= 1))) = 0);
```

WR9: NOT(is_assembly_module_usage(SELF))
OR

```
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'maximum negative component height')) <= 1))) = 0);
```

WR10: NOT(is_assembly_module_usage(SELF))
OR

```
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'maximum positive component height')) <= 1))) = 0);
```

WR11: NOT(is_assembly_module_usage(SELF))
OR

```
(SIZEOF (QUERY (pdr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') |
(pdr\product_definition_relationship.name = 'implemented function') AND
(pdr.relateing_product_definition.frame_of_reference.name =
'functional design usage')) = 1);
```

```

WR12: NOT(is_assembly_module_design(SELF))
      OR
      (SIZEOF (QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATING_PRODUCT_DEFINITION') |
        (is_laminate_component(pdr.related_product_definition)
        ))) = 0));

WR14: NOT(is_interconnect_module_usage(SELF))
      OR
      (SIZEOF (QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') |
        (pdr\product_definition_relationship.name = 'implemented function') AND
        (pdr.relying_product_definition.frame_of_reference.name =
        'functional design usage')))) <= 1));

WR15: NOT(is_interconnect_module_usage(SELF))
      OR
      (SIZEOF (QUERY (pd1 <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_SHAPE' IN TYPEOF (pd1))
        AND (SIZEOF(QUERY(cd <* USEDIN(pd1,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CHARACTERIZED_DEFINITION' IN TYPEOF (cd))
        AND (SIZEOF(QUERY(pd2 <* USEDIN(cd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        (pd2\property_definition.name = 'located interconnect module thickness')
        AND (SIZEOF(USEDIN(pd2,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) > 1)
        )) = 1))) = 1))) <=1 ));

WR16: NOT(is_interconnect_module_usage(SELF))
      OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        (pd\property_definition.name =
        'interconnect module usage view physical characteristics')))) <= 1);

```

```

WR17: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
(pd\property_definition.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(pdr.used_representation\representation.name =
'minimum thickness over metal requirement')
AND ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF (pdr.used_representation))
)) <= 1))) <= 1);

```

```

WR18: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
(pd\property_definition.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(pdr.used_representation\representation.name =
'maximum thickness over metal requirement')
AND ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF (pdr.used_representation))
)) <= 1))) <= 1);

```

```

WR19: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
(pd\property_definition.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(pdr.used_representation\representation.name =
'minimum thickness over dielectric requirement')
AND ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF (pdr.used_representation))
)) <= 1))) <= 1);

```

```

WR20: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
(pd\property_definition.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(pdr.used_representation\representation.name =
'maximum thickness over dielectric requirement')
AND ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF (pdr.used_representation))
)) <= 1))) <= 1);

WR21: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
((pd\property_definition.name = 'restraint') OR
(pd\property_definition.name = 'tolerance specific restraint')))) <= 1);

WR22: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
(sa\shape_aspect.description =
'interconnect module primary surface')
)) <= 1))) = 0);

WR23: NOT(is_interconnect_module_usage(SELF))
OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
(sa\shape_aspect.description =
'interconnect module secondary surface')
)) <= 1))) = 0);

```

```

WR24: NOT(is_interconnect_module_usage(SELF))
      OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      (sa\shape_aspect.description =
        'interconnect module edge surface')
      )) <= 1))) = 0));

WR25: NOT(is_assembly_module_design(SELF)) OR
      ((NOT ((SELF.frame_of_reference.name = 'physical design') AND
      (SELF\product_definition.name = 'assembly module')
      AND (SIZEOF (QUERY(pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF (QUERY(sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_JOINT' IN TYPEOF (sa)))) > 0 ))) = 0)))
      OR (SIZEOF (QUERY(pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF (QUERY(sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_JOINT' IN TYPEOF (sa)) AND
      (SIZEOF (QUERY( acu <* USEDIN (
      sa\shape_aspect_relationship.relatng_shape_aspect.of_shape.definition,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (acu))
      AND (acu\product_definition_relationship.relatng_product_definition = SELF)
      )) >= 1))) >=1 ))) >= 1));

```

```

WR26: NOT(is_assembly_module_design(SELF)) OR
((NOT ((SELF.frame_of_reference.name = 'physical design') AND
(SELF\product_definition.name = 'assembly module')
))
AND (SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))) = 0) OR
(SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))AND
(SIZEOF (QUERY( acu <* USEDIN (aj\
shape_aspect_relationship.related_shape_aspect,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF (acu))
AND (acu\product_definition_relationship.relateing_product_definition = SELF)
)) = 0))) = 0));

```

```

WR27: NOT(is_assembly_module_design(SELF)) OR
((NOT ((SELF.frame_of_reference.name = 'physical design') AND
(SELF\product_definition.name = 'assembly module')
))
AND (SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)))) = 0) OR
(SIZEOF (QUERY(aj <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT' IN TYPEOF (aj)) AND
(SIZEOF (QUERY( cl <* USEDIN (aj\
shape_aspect_relationship.related_shape_aspect.of_shape.definition,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REPRESENTATION.ITEMS') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_LOCATION' IN TYPEOF (cl))
AND (aj IN cl\representation.items))) = 0))) = 0));

```



```

WR28: NOT(is_interconnect_module_design(SELF)) OR
      ((NOT ((SELF.frame_of_reference.name = 'physical design') AND
              (SELF\product_definition.name = 'interconnect module')
            )) OR
        (SIZEOF (QUERY (pdr <* USEDIN (SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_DEFINITION_RELATIONSHIP.' +
          'RELATING_PRODUCT_DEFINITION') |
            (
              ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr))
            AND
              (pdr\product_definition_relationship.name = 'interconnect module stratum assembly r
            AND
              (pdr.related_product_definition\product_definition.description = 'primary design lay
            AND
              ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'STRATUM' IN TYPEOF(pdr.related_product_definition))
            )
          )
        ) = 1));

END_ENTITY; -- physical_unit
(*

```

Formal propositions:

WR1: If the physical_unit is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR2: The physical_unit shall have exactly one creation date.

WR3: The physical_unit shall have exactly one approval.

WR4: The physical_unit shall have at least one organization or person_and_organization as the creator.

WR5: The physical_unit shall have exactly one security_classification.

WR6: The physical_unit shall have a frame_of_reference with a name of 'physical design' or 'physical design usage'.

WR7: The physical_unit shall be related to at most one property with a name of 'unit technology'.

WR8: If the physical_unit is a 'physical design usage' of an 'assembly module', it shall have at least one shape_aspect that is of type assembly_module_terminal.

WR9: If the physical_unit is a 'physical design usage' of an 'assembly module', it shall have at most one representation with a name of 'maximum negative component height'

WR10: If the physical_unit is a 'physical design usage' of an 'assembly module', it shall have at most one representation with a name of 'maximum positive component height'.

WR11: If the `physical_unit` is a 'physical design usage' of an 'assembly module', it shall have exactly one 'functional design usage' specified as the implemented function.

WR12: If the `physical_unit` is a 'physical design' of an 'assembly module', it shall have no laminate components.

WR13: If the `physical_unit` is a 'physical design usage' of an 'interconnect module', it shall have no more than one 'functional design usage' specified as the 'implemented function'.

WR14: If the `physical_unit` is a 'physical design usage' of an 'interconnect module', it shall have at most one `requirements_property` with a name of 'located interconnect module thickness'

WR15: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'interconnect module usage view physical characteristics'.

WR16: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'interconnect module usage view physical characteristics' and related to a `length_measure_with_unit` with a name of 'minimum thickness over metal requirement'

WR17: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'interconnect module usage view physical characteristics' and related to a `length_measure_with_unit` with a name of 'maximum thickness over metal requirement'

WR18: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'interconnect module usage view physical characteristics' and related to a `length_measure_with_unit` with a name of 'minimum thickness over dielectric requirement'

WR19: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'interconnect module usage view physical characteristics' and related to a `length_measure_with_unit` with a name of 'maximum thickness over dielectric requirement'

WR20: If the `physical_unit` is a physical design usage of an interconnect module it shall be the definition for at most one `property_definition` with a name of 'restraint' or 'tolerance specific restraint'.

WR21: If the `physical_unit` is a 'physical design usage' of an 'interconnect module', it shall have no more than one `shape_aspect` that is implementation of secondary surface.

WR22: If the `physical_unit` is a 'physical design usage' of an 'interconnect module', it shall have no more than one `shape_aspect` that is implementation of primary surface.

WR23: If the `physical_unit` is a 'physical design usage' of an 'interconnect module', it shall have no more than one `shape_aspect` that is implementation of edge surface.

WR24: If the `physical_unit` that is a design view of an assembly module is referenced by an `assembly_joint` through the `of_shape` attribute, at least one of the members of `component_definition` that are related by the `relating_shape_aspect` of that `assembly_joint` shall be related by members of `assembly_component_usage` to this `physical_unit`.

WR25: If the `physical_unit` that is a design view of an assembly module is referenced by an `assembly_joint` through the `of_shape` attribute, each member of `component_definition` that are related by the `related_shape_aspect` of that `assembly_joint` shall be related by members of `assembly_component_usage` to this `physical_unit`.

WR26: If the `physical_unit` that is a design view of an assembly module is referenced by an `assembly_joint` through the `of_shape` attribute, the `component_definition` that is related by the `assembly_joint\shape_aspect_relationship.related_shape_aspect` attribute of that `assembly_joint` shall be positioned by a member of `component_location` which also references that `assembly_joint`.

WR27: If the `physical_unit` that is a design view of an interconnect module, there shall be exactly one primary design layer stratum related to the `physical_unit` by an `assembly_component_usage`.

5.2.3.1.270 `physical_unit_datum`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_unit_datum` is a `shape_aspect` that implements the ARM concepts of:

- `Common_datum` (see 4.2.105);
- `Datum` (see 4.2.214);
- `Datum_axis` (see 4.2.215);
- `Datum_plane` (see 4.2.232);
- `Datum_point` (see 4.2.233);
- `Edge_segment_vertex` (see 4.2.312);
- `Single_datum` (see 4.2.828).

EXPRESS specification:

```
*)
ENTITY physical_unit_datum
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\shape_aspect.description IN ['axis', 'plane', 'point', ''];

  WR2: SELF\shape_aspect.product_definitional = False;
```

```

WR3: SELF\shape_aspect.name IN ['single datum', 'common datum', ''];

WR4: (NOT (SELF\shape_aspect.name = 'common datum')) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      (sar\shape_aspect_relationship.name = 'datum feature usage in datum system') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_SYSTEM' IN TYPEOF(SAR)))) >= 1);

WR5: (NOT (SELF\shape_aspect.name = 'common datum')) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      (sar\shape_aspect_relationship.name = 'datum feature usage in common datum') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_SYSTEM' IN TYPEOF(SAR)))) >= 2);

WR6: (NOT ((SELF\shape_aspect.description = 'axis') AND
      (SELF\shape_aspect.name IN ['common datum', 'single datum']))) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      NOT(sar\shape_aspect_relationship.name = 'reference axis') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_REFERENCE_FRAME' IN TYPEOF(SAR)))) = 1);

WR7: (NOT ((SELF\shape_aspect.description = 'plane') AND
      (SELF\shape_aspect.name IN ['common datum', 'single datum']))) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      NOT(sar\shape_aspect_relationship.name = 'reference plane') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_REFERENCE_FRAME' IN TYPEOF(SAR)))) = 1);

WR8: (NOT ((SELF\shape_aspect.description = 'point') AND
      (SELF\shape_aspect.name IN ['common datum', 'single datum']))) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      NOT(sar\shape_aspect_relationship.name = 'reference origin') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_REFERENCE_FRAME' IN TYPEOF(SAR)))) = 1);

```

```

WR9: (NOT (SELF\shape_aspect.name = 'single datum')) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      (sar\shape_aspect_relationship.name = 'datum usage in datum system') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_SYSTEM' IN TYPEOF(SAR)))) >= 1);

WR10: (NOT (SELF\shape_aspect.name = 'single datum')) OR
      (SIZEOF (QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      (sar\shape_aspect_relationship.name = 'datum feature usage in single datum') AND
      (TRUE))) <= 1);

WR11: NOT ((SELF\shape_aspect.name IN [''])
      AND (SELF\shape_aspect.description IN ['']));

WR12: (NOT(SELF\shape_aspect.description = 'plane')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      pd\property_definition.description = 'datum direction property')) <= 2);
END_ENTITY; -- physical_unit_datum
(*)

```

Formal propositions:

- WR1:** The physical_unit_datum shall have a description of 'axis', 'plane', 'point' or 'null'.
- WR2:** The physical_unit_datum shall not lie on the boundary of the shape of the product.
- WR3:** The physical_unit_datum shall have a name of 'single datum', 'common datum', or 'null'.
- WR4:** If the name of the physical_unit_datum is 'common datum' it shall be related to at least one datum_system as a 'datum usage in datum system'.
- WR5:** If the name of the physical_unit_datum is 'common datum', it shall relate to at least two shape_aspect to specify the 'datum feature usage in common datum'.
- WR6:** If the description of the physical_unit_datum is 'axis' and the physical_unit_datum is not related to a datum_reference_frame as a 'reference axis', it shall have a name of either 'common datum' or 'single datum'.
- WR7:** If the description of the physical_unit_datum is 'plane' and the physical_unit_datum is not related to a datum_reference_frame as a 'reference plane', it shall have a name of either 'common datum' or 'single datum'.

WR8: If the description of the `physical_unit_datum` is 'point' and the `physical_unit_datum` is not related to a `datum_reference_frame` as a 'reference origin', it shall have a name of either 'common datum' or 'single datum'.

WR9: If the name of the `physical_unit_datum` is 'single datum', it shall be related to at least one `datum_system` as a 'datum usage in datum system'.

WR10: If the name of the `physical_unit_datum` is 'single datum', it shall relate to no more than one `shape_aspect` to specify the 'datum feature usage in single datum'.

WR11: The name and description of the `physical_unit_datum` shall not both be an empty string.

WR12: If the description of the `physical_unit_datum` is 'plane', it shall have at most two `property_definitions` with the description of 'datum direction property'.

Informal propositions:

IP1: If the description of the `physical_unit_datum` is 'plane' and it has two `property_definitions` with the description of 'datum direction property', the names of the two `property_definitions` shall not be the same.

5.2.3.1.271 `physical_unit_datum_feature`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_unit_datum_feature` is a `shape_aspect` that implements the ARM concept of `Datum_feature` (see 4.2.226).

EXPRESS specification:

```
*)
ENTITY physical_unit_datum_feature
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.' + 'DEFINITION') |
    (pd\property_definition.description = 'datum feature identification')))) = 1;

  WR2: (SIZEOF (QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name IN ['datum feature usage in common datum',
    'datum feature usage in single datum'])) >= 1);
```

```

WR3: (SIZEOF (QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'datum feature usage in single datum')) <= 1
END_ENTITY; -- physical_unit_datum_feature
(*

```

Formal propositions:

WR1: The `physical_unit_datum_feature` shall have exactly one property that has the description of 'datum feature identification'.

WR2: The `physical_unit_datum_feature` shall be related by at least one `shape_aspect_relationship` with a name of either 'datum feature usage in common datum' or 'datum feature usage in single datum'.

WR3: The `physical_unit_datum_feature` shall be related to at most one `shape_aspect` as a 'datum feature usage in single datum'.

Informal propositions:

IP1: The `physical_unit_datum_feature` shall either lie on the boundary of the shape of the product or be a `part_template_definition`.

5.2.3.1.272 `physical_unit_datum_target`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_unit_datum_target` is a `shape_aspect` that implements that ARM concept of `Datum_target` (see 4.2.240).

EXPRESS specification:

```

*)
ENTITY physical_unit_datum_target
    SUBTYPE OF (shape_aspect);
WHERE

    WR1: (SIZEOF (QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'datum target usage')) >= 1);
END_ENTITY; -- physical_unit_datum_target
(*

```

Formal propositions:

WR1: The physical_unit_datum_target shall be related to at least one shape_aspect as the 'datum target usage'.

5.2.3.1.273 physical_unit_datum_target_set

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A physical_unit_datum_target_set is a physical_unit_datum_feature that implements the ARM concept of Datum_target_set (see 4.2.241).

EXPRESS specification:

```

*)
ENTITY physical_unit_datum_target_set
  SUBTYPE OF (physical_unit_datum_feature);
WHERE

  WR1: (SIZEOF (QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'datum target usage')) >= 1);

  WR2: (SIZEOF (QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'constituent')) = 0);
END_ENTITY; -- physical_unit_datum_target_set
( *
```

Formal propositions:

WR1: The physical_unit_datum_target_set shall relate to at least one shape_aspect to specify the 'datum target usage'.

WR2: The physical_unit_datum_target_set shall not be related to a shape_aspect as a 'constituent'.

5.2.3.1.274 physical_unit_geometric_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `physical_unit_geometric_tolerance` is a `geometric_tolerance` and a `property_definition` that implements the ARM concepts of:

- `Circularity_tolerance` (see 4.2.103);
- `Flatness_tolerance` (see 4.2.386);
- `Geometric_tolerance` (see 4.2.419);
- `Geometric_tolerance_without_referenced_datum_system` (see 4.2.423);
- `Straightness_tolerance` (see 4.2.844).

EXPRESS specification:

```

*)
ENTITY physical_unit_geometric_tolerance
  SUBTYPE OF (geometric_tolerance, property_definition);
WHERE

  WR1: SELF\geometric_tolerance.name = SELF\property_definition.name;

  WR2: SELF\geometric_tolerance.toleranced_shape_aspect
        = SELF\property_definition.definition;

  WR3: (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_PROFILE_TOLERANCE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SURFACE_PROFILE_TOLERANCE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE'] * TYPEOF(SELF)) = 0))
    OR (SELF\geometric_tolerance.name IN ['circularity',
        'cylindricity', 'flatness', 'straightness']));

  WR4: (NOT (SELF\geometric_tolerance.name = 'circularity')) OR
        (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF)));

  WR5: (NOT (SELF\geometric_tolerance.name = 'cylindricity')) OR
        (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF)));

```

```

WR6: (NOT (SELF\geometric_tolerance.name = 'flatness')) OR
      (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF)));

WR7: (SIZEOF (QUERY (pugt <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
            sar\shape_aspect_relationship.name = 'group geometric tolerance') |
            pugt.relate_shape_aspect\shape_aspect.description =
            'simultaneous requirement')) <= 1);
END_ENTITY; -- physical_unit_geometric_tolerance
( *

```

Formal propositions:

WR1: The value of two name attributes of the `physical_unit_geometric_tolerance` shall be identical.

WR2: The `shape_aspect` specified as the definition of the `physical_unit_geometric_tolerance` shall be the same `shape_aspect` specified as the `toleranced_shape_aspect` by the `physical_unit_geometric_tolerance`.

WR3: If the `physical_unit_geometric_tolerance` is neither a `geometric_tolerance_with_specified_datum_system`, a `linear_profile_tolerance`, a `surface_profile_tolerance`, nor a `position_tolerance`, the name of the `physical_unit_geometric_tolerance` shall be either 'circularity', 'cylindricity', 'flatness', or 'straightness'.

WR4: If the name of the `physical_unit_geometric_tolerance` is 'circularity', it shall not be a `modified_geometric_tolerance`.

WR5: If the name of the `physical_unit_geometric_tolerance` is 'cylindricity', it shall not be a `modified_geometric_tolerance`.

WR6: If the name of the `physical_unit_geometric_tolerance` is 'flatness', it shall not be a `modified_geometric_tolerance`.

WR7: The `physical_unit_geometric_tolerance` shall be related to at most one `geometric_tolerance_group` with a description of 'simultaneous requirement' as the 'group geometric tolerance'.

Informal propositions:

IP1: The `physical_unit_geometric_tolerance` either shall relate to exactly one `property_definition` that has a description of either 'conical tolerance zone boundary size', 'positional boundary member property', 'profile related positional boundary property' to specify the 'defined boundary'; or shall be the `defining_tolerance` of exactly one `tolerance_zone` that is the zone of exactly one `tolerance_zone_definition`.

IP2: The `toleranced_shape_aspect` of the `physical_unit_geometric_tolerance` shall either lie on the boundary of the shape of the product; be a `centre_of_symmetry` with a description of either 'axis', 'plane', or 'point'; be a tangent with a description of 'plane'; be a `part_template_definition`; or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which either are all of type `part_template_definition`, are all of type tangent with a description of 'plane', or are all of type `centre_of_symmetry` and have the same description – either 'axis', 'plane', or 'point'.

IP3: If the name of the `physical_unit_geometric_tolerance` is 'circularity', the `toleranced_shape_aspect` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP4: If the name of the `physical_unit_geometric_tolerance` is 'circularity', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

IP5: If the name of the `physical_unit_geometric_tolerance` is 'cylindricity', the `toleranced_shape_aspect` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP6: If the name of the `physical_unit_geometric_tolerance` is 'cylindricity', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

IP7: If the name of the `physical_unit_geometric_tolerance` is 'flatness', the `toleranced_shape_aspect` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP8: If the name of the `physical_unit_geometric_tolerance` is 'flatness', the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

IP9: If the name of the `physical_unit_geometric_tolerance` is 'flatness' and the `physical_unit_geometric_tolerance` is the defining `tolerance` of a `tolerance_zone` that has a `property_definition` with a description of 'tolerance zone per unit size', the `property_definition` shall have exactly one representation that contains two `length_measure_with_unit` with a name of 'per unit size'.

IP10: There shall exist exactly one `requirements_property` that has a representation that contains a `descriptive_representation_item` that has the name of 'requirements description' and a description of 'geometric dimensioning and tolerancing reference', and that `requirements_property` shall be an item of one or more `document_reference` the assigned `document` of which are `ee_specification`.

IP11: If the name of the `physical_unit_geometric_tolerance` is 'straightness'; the `toleranced_shape_aspect` shall either lie on the boundary of the shape of the product; be a `centre_of_symmetry` with a description of either 'axis' or 'plane', be a `part_template_definition`; or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which either are all of type `part_template_definition`, or are all of type `centre_of_symmetry` and have the same description – either 'axis' or 'plane'.

IP12: If the name of the `physical_unit_geometric_tolerance` is 'straightness' and the `toleranced_shape_-aspect` of the `physical_unit_geometric_tolerance` is either a `centre_of_symmetry` with a description of 'axis', or a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `centre_of_symmetry` and have a description of 'axis'; the `tolerance_zone_boundary` shall have a description of 'circular or cylindrical or spherical'.

IP13: If the name of the `physical_unit_geometric_tolerance` is 'straightness' and the `toleranced_shape_-aspect` of the `physical_unit_geometric_tolerance` is neither a `centre_of_symmetry` with a description of 'axis', nor a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `centre_of_symmetry` and have a description of 'axis'; the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

IP14: If the name of the `physical_unit_geometric_tolerance` is 'straightness' and the `toleranced_shape_-aspect` either is a `part_template_definition`, lies on the boundary of the shape of the product, or is a `composite_shape_aspect` in which all the 'composing' `shape_aspects` are of type `part_template_definition`; the `physical_unit_geometric_tolerance` shall be the `defining_tolerance` of a `tolerance_zone` that has exactly one `property_definition` with a description of 'tolerance zone orientation' that relates to a `property_definition` with a description of 'viewing plane based orientation' to specify its 'zone orientation'.

IP15: If the name of the `physical_unit_geometric_tolerance` is 'straightness' and the `toleranced_shape_-aspect` is not a `part_template_definition`, does not lie on the boundary of the shape of the product, or is not a `composite_shape_aspect` in which all the 'composing' `shape_aspects` are of type `part_template_definition`; the `physical_unit_geometric_tolerance` shall be the `defining_tolerance` of a `tolerance_zone` that has exactly zero `property_definition` with a description of 'tolerance zone orientation'.

5.2.3.1.275 `physical_unit_network_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `physical_unit_network_definition` is a `product_definition` that implements the ARM concept of `Physical_unit_network_definition` (see 4.2.668)

EXPRESS specification:

```
*)
ENTITY physical_unit_network_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF (SELF))) OR
    (SIZEOF (QUERY (docs <*
    SELF\product_definition_with_associated_documents.
    documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);
```

```

WR2: sizeof (QUERY (adta <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) +
    sizeof (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ada.role\date_role.name = 'creation date')) = 1;

WR3: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

WR4: sizeof (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    sizeof (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

WR5: sizeof (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

WR6: sizeof (typeof (SELF) - (typeof (SELF\product_definition ||
    SELF\physical_unit_network_definition))) = 0;

WR7: SELF.frame_of_reference.name IN ['physical network design'];
END_ENTITY; -- physical_unit_network_definition
(*)

```

Formal propositions:

WR1: If the `physical_unit_network_definition` is a `product_definition_with_associated_documents`, there shall be at most one document in the set of `documentation_ids` that has a `product_data_type` of 'CAD filename'

WR2: The `physical_unit_network_definition` shall have exactly one creation date.

WR3: The `physical_unit_network_definition` shall have exactly one approval.

WR4: The `physical_unit_network_definition` shall have at least one organization or `person_and_organization` as the creator.

WR5: The `physical_unit_network_definition` shall have exactly one `security_classification`.

WR6: The `physical_unit_network_definition` shall not be any other type of `product_definition`.

WR7: The `physical_unit_network_definition` shall have a `frame_of_reference` with a name of 'physical network design'.

5.2.3.1.276 `plated_cutout_edge_segment`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `plated_cutout_edge_segment` is a `cutout_edge_segment` and `plated_inter_stratum_feature` that implements the ARM concept of `Plated_cutout_edge_segment` (see 4.2.686).

EXPRESS specification:

```
*)
ENTITY plated_cutout_edge_segment
  SUBTYPE OF (cutout_edge_segment, plated_inter_stratum_feature);
WHERE

  WR1: SELF\shape_aspect.description IN ['plated cutout edge segment'];

  WR2: SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'join implementation') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT' IN
    TYPEOF (ji.relate_shape_aspect)) AND
    (ji.relate_shape_aspect\shape_aspect.name =
    'inter stratum join')))) <= 1;

  WR3: (NOT (SELF\shape_aspect.description = 'plated cutout edge segment')) OR
    (SIZEOF (QUERY (cc <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'composed cutout') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLATED_INTER_STRATUM_FEATURE' IN
    TYPEOF (cc.relate_shape_aspect)) AND
    (cc.relate_shape_aspect\shape_aspect.description =
    'plated cutout')))) = 1);
END_ENTITY; -- plated_cutout_edge_segment
(*
```

Formal propositions:

WR1: The `plated_cutout_edge_segment` shall have a description of 'plated cutout edge segment'.

WR2: The `plated_cutout_edge_segment` shall be related to at most one `join_shape_aspect` with a name of 'inter stratum join' as a 'join implementation'.

WR3: The `plated_cutout_edge_segment` shall be related to exactly one cutout with a description of 'plated cutout' as the 'composed cutout'.

5.2.3.1.277 **plated_inter_stratum_feature**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `plated_inter_stratum_feature` is an `inter_stratum_feature` that implements the ARM concepts of:

- `Inter_stratum_join_implementation` (see 4.2.436);
- `Join_two_physical_connectivity_definition_supporting_inter_stratum_feature` (see 4.2.480);
- `Physical_network_supporting_inter_stratum_feature` (see 4.2.661);
- `Plated_cutout` (see 4.2.685);
- `Plated_inter_stratum_feature` (see 4.2.687);
- `Plated_interconnect_module_edge` (see 4.2.688).

EXPRESS specification:

```

*)
ENTITY plated_inter_stratum_feature
  SUPERTYPE OF (ONEOF(plated_passage,plated_cutout_edge_segment,
    plated_interconnect_module_edge_segment))
  SUBTYPE OF (inter_stratum_feature);
WHERE

  WR1: (SELF\shape_aspect.description IN ['bonded conductive base blind via',
    'buried via',
    'component termination passage', 'interfacial connection',
    'non conductive base blind via',
    'join two physical connectivity definition supporting inter stratum feature',
    'plated conductive base blind via',
    'plated cutout', 'plated interconnect module edge']) OR
    (SIZEOF
      ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_CUTOUT_EDGE_SEGMENT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_INTERCONNECT_MODULE_EDGE_SEGMENT'] * TYPEOF (SELF)) = 1 );

```

```

WR2: SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'join implementation') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT' IN
    TYPEOF (ji.relatng_shape_aspect)) AND
    (ji.relatng_shape_aspect\shape_aspect.name =
    'inter stratum join')))) <= 1;
END_ENTITY; -- plated_inter_stratum_feature
(*

```

Formal propositions:

WR1: The `plated_inter_stratum_feature` shall have a description of either 'bonded conductive base blind via', 'buried via', 'component termination passage', 'interfacial connection', 'non conductive base blind via', 'plated conductive base blind via', 'plated cutout', 'join two physical connectivity definition supporting inter stratum feature', or 'plated interconnect module edge' or shall be a segment subtype.

WR2: The `plated_inter_stratum_feature` shall be related to at most one `join_shape_aspect` with a name of 'inter stratum join' as a 'join implementation'.

Informal propositions:

IP1: The `plated_inter_stratum_feature` shall have an associated `passage_technology` that shall have descriptive_representation_item with a description of 'true'.

IP2: If the `plated_inter_stratum_feature` description is 'join two physical connectivity definition supporting inter stratum feature', it shall be associated with two physical connectivity definitions as the joined definitions through `property_definition_relationship` instances.

5.2.3.1.278 **plated_interconnect_module_edge_segment**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `plated_interconnect_module_edge_segment` is a `plated_inter_stratum_feature` and `interconnect_module_edge_segment` that implements the ARM concept of `Plated_interconnect_module_edge_segment` (see 4.2.689).

EXPRESS specification:

```

*)
ENTITY plated_interconnect_module_edge_segment
  SUBTYPE OF (interconnect_module_edge_segment,
              plated_inter_stratum_feature);
WHERE

  WR1: SELF\shape_aspect.description IN ['plated interconnect module edge segment'];

  WR2: SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'join implementation') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT' IN
    TYPEOF (ji.relate_shape_aspect)) AND
    (ji.relate_shape_aspect\shape_aspect.name =
    'inter stratum join')))) <= 1;

  WR3: (NOT (SELF\shape_aspect.description =
    'plated interconnect module edge segment')) OR
    (SIZEOF (QUERY (ce <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'composed edge') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLATED_INTER_STRATUM_FEATURE' IN
    TYPEOF (ce.relate_shape_aspect)) AND
    (ce.relate_shape_aspect\shape_aspect.description =
    'plated interconnect module edge')))) = 1);
END_ENTITY; -- plated_interconnect_module_edge_segment
(

```

Formal propositions:

WR1: The plated_interconnect_module_edge_segment shall have a description of 'plated interconnect module edge segment'.

WR2: The plated_interconnect_module_edge_segment shall be related to at most one join_shape_aspect with a name of 'inter stratum join' as a 'join implementation'.

WR3: The plated_interconnect_module_edge_segment shall be related to exactly one interconnect_module_edge with a description of 'plated interconnect module edge' as the 'composed edge'.

Informal propositions:

IP1: The associated passage_technology shall have descriptive_representation_item with a description of 'true'.

5.2.3.1.279 plated_passage

A plated_passage is a plated_inter_stratum_feature that implements the ARM concepts of:

- Blind_via (see 4.2.71);
- Bonded_conductive_base_blind_via (see 4.2.74);
- Buried_via (see 4.2.81);
- Component_termination_passage (see 4.2.143);
- Interfacial_connection (see 4.2.474);
- Non_conductive_base_blind_via (see 4.2.557);
- Plated_conductive_base_blind_via (see 4.2.684);
- Plated_passage (see 4.2.690);
- Via (see 4.2.947).

EXPRESS specification:

```
*)  
ENTITY plated_passage  
  SUBTYPE OF (plated_inter_stratum_feature);  
WHERE  
  
  WR1: SELF\shape_aspect.description IN ['bonded conductive base blind via',  
    'buried via', 'component termination passage',  
    'interfacial connection', 'non conductive base blind via',  
    'plated conductive base blind via'];
```

```

WR2: (NOT (SELF\shape_aspect.description = 'bonded conductive base blind via')) OR
(SIZEOF (QUERY (fj <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'features join') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_CONCEPT_RELATIONSHIP' IN
TYPEOF (fj.relateing_shape_aspect)) AND
(fj.relateing_shape_aspect\shape_aspect.name =
'stratum feature conductive join')))) = 1);

WR3: (NOT (SELF\shape_aspect.description = 'component termination passage')) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'instantiated template') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION' IN
TYPEOF (it.relateing_shape_aspect)) AND
(it.relateing_shape_aspect\shape_aspect.description =
'component termination passage template')))) = 1);
( *

WR4: (NOT (SELF\shape_aspect.description = 'component termination passage')) OR
(SIZEOF (QUERY (at <* associated_terminals(SELF) |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT' IN
TYPEOF (at)) AND
(at\shape_aspect.description =
'component termination passage interface terminal')))) >= 0);

WR5: (NOT (SELF\shape_aspect.description = 'component termination passage')) OR
(SIZEOF (QUERY (at <* associated_terminals(SELF) |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL' IN
TYPEOF (at)) AND
(at\shape_aspect.description =
'component termination passage join terminal')))) >= 1);
*)

```

```

WR6: (NOT (SELF\shape_aspect.description IN
    ['bonded conductive base blind via',
     'buried via',
     'interfacial connection',
     'non conductive base blind via',
     'plated conductive base blind via'])) OR
(SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated template') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN
    TYPEOF (it.relatng_shape_aspect)) AND
    (it.relatng_shape_aspect\shape_aspect.description =
    'via template')))) = 1);
END_ENTITY; -- plated_passage
(*

```

Formal propositions:

WR1: The plated_passage shall have a description of 'bonded conductive base blind via', 'buried via', 'component termination passage', 'interfacial connection', 'non conductive base blind via', or 'plated conductive base blind via'.

WR2: If the plated_passage has a description of 'bonded conductive base blind via', the plated_passage shall be related to exactly one stratum_concept_relationship with a name of 'stratum feature conductive join' as a 'features join'.

WR3: If the plated_passage has a description of 'component termination passage', the plated_passage shall be related to exactly one part_template_definition with a description of 'component termination passage template' as an 'instantiated template'.

WR4: If the plated_passage has a description of 'component termination passage', the plated_passage shall relate to zero or more component_shape_aspect that has a description of 'component termination passage interface terminal' to specify the 'associated component'.

WR5: If the plated_passage has a description of 'component termination passage', the plated_passage shall relate to at least one component_terminal that has a description of 'component termination passage join terminal' to specify the 'associated component'.

WR6: If the plated_passage has a description of 'bonded conductive base blind via', 'buried via', 'interfacial connection', 'non conductive base blind via', or 'plated conductive base blind via', the plated_passage shall be related to exactly one part_template_definition with a description of 'via template' as an 'instantiated template'.

5.2.3.1.280 position_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A position_tolerance is a physical_unit_geometric_tolerance that implements the ARM concepts of:

- Boundary_based_position_tolerance (see 4.2.77);
- Composite_feature_locating_position_tolerance (see 4.2.157);
- Composite_feature_relating_position_tolerance (see 4.2.158);
- Position_tolerance (see 4.2.695);
- Position_tolerance_with_referenced_datum_system (see 4.2.696);
- Position_tolerance_without_referenced_datum_system (see 4.2.697).

EXPRESS specification:

*)

ENTITY position_tolerance

SUBTYPE OF (physical_unit_geometric_tolerance);

WHERE

WR1: (NOT(SELF\geometric_tolerance.name = 'boundary based position')) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF));

WR2: (NOT(SELF\geometric_tolerance.name = 'boundary based position')) OR
(SIZEOF(QUERY(gtr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM'
IN TYPEOF(gtr.related_geometric_tolerance)) AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE'
IN TYPEOF(gtr.related_geometric_tolerance)) AND
(gtr.related_geometric_tolerance\geometric_tolerance.name IN ['surface profile',
'surface profile locating', 'surface profile refinement']) AND
(gtr\geometric_tolerance_relationship.name = 'profile control')
)) <= 1);

```

WR3: (NOT(SELF\geometric_tolerance.name =
        'feature relating position')) OR
      (SIZEOF(QUERY(gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
        (gtr\geometric_tolerance_relationship.name = 'position refining control') AND
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE'
        IN TYPEOF(gtr.relatng_geometric_tolerance)) AND
        (gtr.relatng_geometric_tolerance\geometric_tolerance.name IN
        ['feature locating position', 'feature relating position'])
        )) = 1);

WR4: (NOT(SELF\geometric_tolerance.name =
        'feature locating position')) OR
      (SIZEOF(QUERY(gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE'
        IN TYPEOF(gtr.related_geometric_tolerance)) AND
        (gtr.related_geometric_tolerance\geometric_tolerance.name = 'feature relating posit:
        AND (gtr\geometric_tolerance_relationship.name = 'position refining control')
        )) = 1);

WR5: (NOT(SELF\geometric_tolerance.name =
        'feature relating position')) OR
      (SIZEOF(QUERY(gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE'
        IN TYPEOF(gtr.related_geometric_tolerance)) AND
        (gtr.related_geometric_tolerance\geometric_tolerance.name = 'feature relating posit:
        AND (gtr\geometric_tolerance_relationship.name = 'position refining control')
        )) <= 1);
END_ENTITY; -- position_tolerance
(*

```

Formal propositions:

WR1: If the position_tolerance has a name of 'boundary based position', it shall be a modified_geometric_tolerance.

WR2: If the position_tolerance has a name of 'boundary based position', it shall relate to at most one geometric_tolerance_with_specified_datum_system that is a surface_profile_tolerance with a name of either 'surface profile', 'surface profile locating', or 'surface profile refinement' to specify the 'profile control'.

WR3: If the `position_tolerance` has a name of 'feature relating position', it shall be related as the 'position refining control' to exactly one `position_tolerance` with a name of either 'feature locating position' or 'feature relating position'.

WR4: If the `position_tolerance` has a name of 'feature locating position', it shall relate to exactly one `position_tolerance` with a name of 'feature relating position' to specify the 'position refining control'.

WR5: If the `position_tolerance` has a name of 'feature relating position', it shall relate to at most one `position_tolerance` with a name of 'feature relating position' to specify the 'position refining control'.

Informal propositions:

IP1: If `position_tolerance` is a `geometric_tolerance_with_specified_datum_system`, it shall have a name of either 'position', 'feature relating position', 'feature locating position', or 'boundary based position'; otherwise, it shall have the name of either 'position' or 'feature relating position'.

IP2: If the name of the `position_tolerance` is 'boundary based position', the `toleranced_shape_aspect` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP3: If the `position_tolerance` has a name of 'boundary based position' and it relates to exactly one `geometric_tolerance_with_specified_datum_system` that is a `surface_profile_tolerance` with a name of either 'surface profile', 'surface profile locating', or 'surface profile refinement' to specify the 'profile control', the `position_tolerance` and `surface_profile_tolerance` shall specify the same `shape_aspect` as their `toleranced_shape_aspect`.

IP4: If the `position_tolerance` has a name of 'boundary based position' and it relates to exactly one `geometric_tolerance_with_specified_datum_system` that is a `surface_profile_tolerance` with a name of either 'surface profile', 'surface profile locating', or 'surface profile refinement' to specify the 'profile control', the `physical_unit_datum` of the `datum_system` referenced by the `surface_profile_tolerance` shall also be `physical_unit_datum` of the `datum_system` referenced by the `position_tolerance` and shall be assigned the same precedence.

IP5: If the `position_tolerance` has a name of 'boundary based position' and it relates to exactly one `geometric_tolerance_with_specified_datum_system` that is a `surface_profile_tolerance` with a name of either 'surface profile', 'surface profile locating', or 'surface profile refinement' to specify the 'profile control', the `position_tolerance` shall relate to a `property_definition` that has a description of 'profile related positional boundary property' to specify the 'defined boundary'.

IP6: If the `position_tolerance` has a name of 'boundary based position' and it does not relate to a `geometric_tolerance_with_specified_datum_system` that is a `surface_profile_tolerance` with a name of either 'surface profile', 'surface profile locating', or 'surface profile refinement' to specify the 'profile control', the `position_tolerance` shall relate to a `property_definition` that has a description of 'positional boundary member property' to specify the 'defined boundary'.

IP7: If the name of the `position_tolerance` is 'feature locating position', the `toleranced_shape_aspect` shall be a `composite_group_shape_aspect`.

IP8: If the position_tolerance has a name of 'feature relating position', it shall be related as the 'position refining control' to exactly one position_tolerance with a name of either 'feature locating position' or 'feature relating position'. Both of these position_tolerances shall specify the same shape_aspect as the toleranced_shape_aspect.

IP9: If the position_tolerance is not a geometric_tolerance_with_specified_datum_system and has a name of 'feature relating position', it shall not relate to a position_tolerance with a name of 'feature relating position' to specify the 'position refining control'.

IP10: If the position_tolerance has a name of 'feature relating position' and it is related as the 'position refining control' to exactly one position_tolerance with a name of 'feature relating position', it shall either not reference a datum_system or it shall have less physical_unit_datum in its referenced datum_system than the datum_system referenced by the position_tolerance to which it is related as the 'position refining control'.

IP11: If the position_tolerance has a name of 'feature relating position' and it is related as the 'position refining control' to exactly one position_tolerance with a name of 'feature locating position', it shall either not reference a datum_system or it shall have less than or an equal number of physical_unit_datum in its referenced datum_system than the datum_system referenced by the position_tolerance to which it is related as the 'position refining control'.

IP12: If the position_tolerance has a name of 'feature relating position' and it is a geometric_tolerance_with_specified_datum_system, the physical_unit_datum of its referenced datum_system shall be physical_unit_datum of the datum_system of the position_tolerance to which it is related as the 'position refining control' and shall be assigned the same precedence.

IP13: If the position_tolerance has a name of 'feature relating position' and it is a geometric_tolerance_with_specified_datum_system that relates to a property_definition of a datum_system that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition', the position_tolerance to which it is related as the 'position refining control' shall also be a geometric_tolerance_with_specified_datum_system that relates to a property_definition of a datum_system that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition'.

IP14: If the position_tolerance has a name of 'feature relating position' and it is a geometric_tolerance_with_specified_datum_system that relates to a property_definition of a datum_system that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', the position_tolerance to which it is related as the 'position refining control' shall also be a geometric_tolerance_with_specified_datum_system that relates to a property_definition of a datum_system that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', and each of these property_definition shall relate to property_definition with a description of 'datum feature material condition property' to specify their 'applied material condition property'. These property_definition shall be applied to physical_unit_datum_features in the context of the datum_systems referenced by each of the position_tolerance. The name of the property_definition applied to each of the physical_unit_datum_feature in the context of the datum_system referenced by this position_tolerance shall be the same as the name of the property_definition applied to the same physical_unit_datum_feature in the context of the datum_system referenced by the position_tolerance to which this position_tolerance is related as the 'position refining control'.

IP15: If the `position_tolerance` has a name of 'feature relating position', it shall specify a `measure_with_unit` as its magnitude that is less than the `measure_with_unit` specified as the magnitude of the `position_tolerance` to which it is related as the 'position refining control'.

IP16: If the `position_tolerance` has a name of 'feature relating position' and it is specified as the defining `tolerance` by a `tolerance_zone` that has a description of 'conical', the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also be specified as the defining `tolerance` by a `tolerance_zone` that has a description of 'conical'.

IP17: If the `position_tolerance` has a name of 'feature relating position' and it is specified as the defining `tolerance` by a `tolerance_zone` that is the zone of a `projected_zone_definition`, the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also be specified as the defining `tolerance` by a `tolerance_zone` that is the zone of a `projected_zone_definition`.

IP18: If the `position_tolerance` has a name of 'feature relating position' and it is specified as the defining `tolerance` by a `tolerance_zone` that has a `property_definition` with the description of 'tolerance zone orientation', the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also be specified as the defining `tolerance` by a `tolerance_zone` that has a `property_definition` with the description of 'tolerance zone orientation'.

IP19: If the `position_tolerance` has a name of 'feature relating position' and its `tolerance_zone_boundary` has a description of 'conical', the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also have a `tolerance_zone_boundary` that has a description of 'conical'.

IP20: If the `position_tolerance` has a name of 'feature relating position' and its `tolerance_zone_boundary` has a description of 'circular or cylindrical or spherical', the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also have a `tolerance_zone_boundary` that has a description of 'circular or cylindrical or spherical'.

IP21: If the `position_tolerance` has a name of 'feature relating position' and its `tolerance_zone_boundary` is a `tolerance_zone_explicit_opposing_boundary_set`, the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also have a `tolerance_zone_boundary` that is a `tolerance_zone_explicit_opposing_boundary_set`.

IP22: If the `position_tolerance` has a name of 'feature relating position' and its `tolerance_zone_boundary` is a `tolerance_zone_implicit_opposing_boundary_set`, the `position_tolerance` that is related to this `position_tolerance` as the 'position refining control' shall also have a `tolerance_zone_boundary` that is a `tolerance_zone_implicit_opposing_boundary_set`.

IP23: The `toleranced_shape_aspect` of the `position_tolerance` shall either lie on the boundary of the shape of the product; be a `centre_of_symmetry` with a description of either 'axis', 'plane', or 'point'; be a `part_template_definition`; or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which either are all of type `part_template_definition`, or are all of type `centre_of_symmetry` and have the same description – either 'axis', 'plane', or 'point'.

IP24: If the `toleranced_shape_aspect` of the `position_tolerance` is either a `centre_of_symmetry` with a description of 'plane', or a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'plane'; the `tolerance_zone_boundary` shall be either a `tolerance_zone_explicit_opposing_boundary_set` or a `tolerance_zone_implicit_opposing_boundary_set`.

IP25: If the `toleranced_shape_aspect` of the `position_tolerance` is either a `centre_of_symmetry` with a description of 'point', or a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'point'; the `position_tolerance` shall be specified as the defining `tolerance` by exactly one `tolerance_zone` that is specified as the zone of a `tolerance_zone_definition`.

5.2.3.1.281 `positional_boundary`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `positional_boundary` is a `shape_aspect` that implements the ARM concepts of:

- `Dimension_related_positional_boundary` (see 4.2.295);
- `Positional_boundary` (see 4.2.698);
- `Profile_related_positional_boundary` (see 4.2.750).

EXPRESS specification:

```

*)
ENTITY positional_boundary
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SELF\shape_aspect.description IN
    ['dimension related positional boundary',
     'profile related positional boundary'];

  WR2: SELF\shape_aspect.product_definitional = False;

  WR3: (NOT(SELF\shape_aspect.description =
    'dimension related positional boundary'))
    OR (SIZEOF(QUERY(sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POSITIONAL_BOUNDARY_MEMBER' IN
    TYPEOF(sar.related_shape_aspect))
    AND (sar\shape_aspect_relationship.name = 'boundary member')))) >= 1);

```

```

WR4: (NOT(SELF\shape_aspect.description =
    'dimension related positional boundary'))
OR (SIZEOF(QUERY(sar <*USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
    | (sar.related_shape_aspect.product_definitional = True) AND
      (sar\shape_aspect_relationship.name = 'constrained feature')))) +
    SIZEOF(QUERY(sar <*USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN TYPEOF(sar.related_shape_aspect))
      AND (sar\shape_aspect_relationship.name = 'constrained feature'))
    ) = 1);

WR5: (NOT(SELF\shape_aspect.description =
    'dimension related positional boundary'))
OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')
    | (pd\property_definition.description =
      'dimension related positional boundary property')))) = 1);

WR6: (NOT(SELF\shape_aspect.description =
    'profile related positional boundary'))
OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')
    | (pd\property_definition.description =
      'profile related positional boundary property')))) = 1);

WR7: (NOT(SELF\shape_aspect.description =
    'profile related positional boundary'))
OR (SIZEOF(QUERY(sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
    | ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POSITIONAL_BOUNDARY_MEMBER' IN TYPEOF(sar.related_shape_aspect))
      AND (sar\shape_aspect_relationship.name =
        'positional boundary and profile boundary member')))) = 1);
END_ENTITY; -- positional_boundary
(

```

Formal propositions:

WR1: The `positional_boundary` shall have a description of either 'dimension related positional boundary' or 'profile related positional boundary'.

WR2: The `positional_boundary` shall not lie on the boundary of the shape of the product.

WR3: If the description of the `positional_boundary` is 'dimension related positional boundary', the `positional_boundary` shall relate to at least one `positional_boundary_member` to specify the 'boundary member'.

WR4: If the description of the `positional_boundary` is 'dimension related positional boundary', the `positional_boundary` shall either relate to exactly one `shape_aspect` that lies on the boundary of the shape of the product to specify the 'constrained feature' or relate to exactly one `part_template_definition` to specify the 'constrained feature'.

WR5: If the description of the `positional_boundary` is 'dimension related positional boundary', the `positional_boundary` shall have exactly one `property_definition` that has the description of 'dimension related positional boundary property'.

WR6: If the description of the `positional_boundary` is 'profile related positional boundary', the `positional_boundary` shall have exactly one `property_definition` that has the description of 'profile related positional boundary property'.

WR7: If the description of the `positional_boundary` is 'profile related positional boundary' it shall relate to exactly one `shape_aspect` to specify the 'positional boundary and profile boundary member'.

5.2.3.1.282 `positional_boundary_member`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `positional_boundary_member` is a `shape_aspect` that implements the ARM concept of `Positional_boundary_member` (see 4.2.701).

EXPRESS specification:

*)

ENTITY `positional_boundary_member`

SUBTYPE OF (`shape_aspect`);

WHERE

WR1: SELF\shape_aspect.product_definitional = False;

```

WR2: SIZEOF(QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION')
    | (pd\property_definition.description = 'positional boundary member property'))
    ) = 1;

WR3: SIZEOF(QUERY(sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
    | (sar\shape_aspect_relationship.name = 'boundary member') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'POSITIONAL_BOUNDARY' IN TYPEOF(sar.relate_shape_aspect))
      AND (sar.relate_shape_aspect\shape_aspect.description =
      'dimension related positional boundary')))) = 1;
END_ENTITY; -- positional_boundary_member
(*)

```

Formal propositions:

WR1: The positional_boundary_member shall not lie on the boundary of the shape of the product.

WR2: The positional_boundary_member shall have exactly one property_definition that has the description of 'positional boundary member property'.

WR3: The positional_boundary_member shall be related to exactly one positional_boundary as the 'boundary member', and that positional_boundary shall have a description of 'dimension related positional boundary'.

5.2.3.1.283 power_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A power_measure_with_unit is a measure_with_unit that implements the ARM concept of Power_measure_with_unit (see ??).

EXPRESS specification:

```

*)
ENTITY power_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
  SELF\power_measure_with_unit))) = 0;

WR2: SELF\measure_with_unit.unit_component\si_unit.name = watt;

```

```

    WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
        TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- power_measure_with_unit
(*

```

Formal propositions:

WR1: The power_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The power_measure_with_unit shall reference an si_unit with a name of watt.

WR3: The unit_component shall be an si_unit.

5.2.3.1.284 pre_defined_parallel_datum_axis_symbol_3d_2d_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An pre_defined_parallel_datum_axis_symbol_3d_2d_relationship is a volume_shape_intersection that implements the ARM concept of Pre_defined_parallel_datum_axis_symbol_3d_2d_relationship (see 4.2.710).

EXPRESS specification:

```

*)
ENTITY pre_defined_parallel_datum_axis_symbol_3d_2d_relationship
    SUBTYPE OF (volume_shape_intersection);
WHERE

    WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
        SELF\representation_relationship ||
        SELF\shape_representation_relationship ||
        SELF\volume_shape_intersection ||
        SELF\pre_defined_parallel_datum_axis_symbol_3d_2d_relationship))) = 0;
END_ENTITY; -- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship
(*

```

Formal propositions:

WR1: The pre_defined_parallel_datum_axis_symbol_3d_2d_relationship shall not be a complex subtype of other than its direct supertypes.

5.2.3.1.285 pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship` is a `volume_shape_intersection` that implements the ARM concept of `Pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship` (see 4.2.712).

EXPRESS specification:

```

*)
ENTITY pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
    SELF\representation_relationship ||
    SELF\shape_representation_relationship ||
    SELF\volume_shape_intersection ||
    SELF\pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship))) = 0;
END_ENTITY; -- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship
( *

```

Formal propositions:

WR1: The `pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship` shall not be a complex subtype of other than its direct supertypes.

5.2.3.1.286 `pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship` is a `volume_shape_intersection` that implements the ARM concept of `Pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship` (see 4.2.714).

EXPRESS specification:

```

*)
ENTITY pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
WHERE

```

```

    WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
        SELF\representation_relationship ||
        SELF\shape_representation_relationship ||
        SELF\volume_shape_intersection ||
        SELF\pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship))) = 0;
END_ENTITY; -- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship
(*)

```

Formal propositions:

WR1: The `pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship` shall not be a complex subtype of other than its direct supertypes.

5.2.3.1.287 **pressure_measure_with_unit**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `pressure_measure_with_unit` is a `measure_with_unit` that implements the ARM concept of `Pressure_measure_with_unit` (see ??).

EXPRESS specification:

```

*)
ENTITY pressure_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

    WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
        SELF\pressure_measure_with_unit))) = 0;

    WR2: SELF\measure_with_unit.unit_component\si_unit.name = pascal;

    WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
        TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- pressure_measure_with_unit
(*)

```

Formal propositions:

WR1: The `pressure_measure_with_unit` shall not be any other type of `measure_with_unit`.

WR2: The `pressure_measure_with_unit` shall reference an `si_unit` with a name of `pascal`.

WR3: The `unit_component` shall be an `si_unit`.

5.2.3.1.288 primary_orientation_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A primary_orientation_feature is a physical_unit_datum_feature that implements the ARM concept of Primary_orientation_feature (see 4.2.717).

EXPRESS specification:

```

*)
ENTITY primary_orientation_feature
  SUBTYPE OF (physical_unit_datum_feature);
WHERE

  WR1: SELF\shape_aspect.product_definitional = TRUE;

  WR2: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_BOTTOM_SURFACE' IN
    TYPEOF (it.relying_shape_aspect))
  )) <= 1;

  WR3: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_TOP_SURFACE' IN
    TYPEOF (it.relying_shape_aspect))
  )) <= 1;
END_ENTITY; -- primary_orientation_feature
(

```

Formal propositions:

WR1: A primary_orientation_feature shall be on the surface of the product.

WR2: A primary_orientation_feature shall be associated with no more than one package_body_bottom_surface.

WR3: A primary_orientation_feature shall be associated with no more than one package_body_top_surface.

5.2.3.1.289 primary_reference_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A primary_reference_terminal is a package_terminal that implements the ARM concept of Primary_reference_terminal (see 4.2.718).

EXPRESS specification:

```
*)
ENTITY primary_reference_terminal
  SUBTYPE OF (package_terminal);
WHERE

WR1: SIZEOF(QUERY ( sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'precedent feaure' ))
  = 0;

WR2: SIZEOF(QUERY ( sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'precedent feaure' ))
  <= 1;
END_ENTITY; -- primary_reference_terminal
(*
```

Formal propositions:

WR1: The primary_reference_terminal shall be referenced by no shape_aspect_relationship with a description of 'precedent feature' as the related_shape_aspect.

WR2: The primary_reference_terminal shall be referenced by no more than one shape_aspect_relationship with a description of 'precedent feature' as the relating_shape_aspect.

5.2.3.1.290 printed_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A printed_component is a component_definition that implements the ARM concepts of:

- Layout_macro_component (see 4.2.504);
- Printed_component (see 4.2.721);

— Printed_connector_component (see 4.2.724).

EXPRESS specification:

```

*)
ENTITY printed_component
  SUPERTYPE OF (layout_macro_component)
  SUBTYPE OF (component_definition);
WHERE

  WR1: SELF.frame_of_reference.name = 'layout occurrence';

  WR2: SELF\product_definition.description = 'printed component';

  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pt_occ <* QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    sa\shape_aspect.description = 'part template occurrence') |
    SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (pt_occ,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'instantiated template') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN
    TYPEOF (it.relate_shape_aspect)) AND
    (it.relate_shape_aspect\shape_aspect.description IN ['printed part template',
    'printed part cross section template',
    'printed connector template']))) = 1)) = 1))) = 0;

  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL' IN
    TYPEOF (sa)) AND
    (sa\shape_aspect.description = 'printed component join terminal')))) >= 1))) = 0;

```

```

WR5: (NOT (SELF\product_definition.description = 'printed connector component')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAMINATE_COMPONENT_INTERFACE_TERMINAL' IN
      TYPEOF (sa)) AND
      (sa\shape_aspect.description =
        'printed connector component interface terminal')))) >= 1))) = 0);

WR6: (NOT (SELF\product_definition.description = 'printed connector component')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (pt_occ <* QUERY (sa <* USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      sa\shape_aspect.description = 'part template occurrence') |
      SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (pt_occ,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'instantiated template') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION' IN
      TYPEOF (it.relateing_shape_aspect)) AND
      (it.relateing_shape_aspect\shape_aspect.description =
        'printed connector template')))) = 1)) = 1))) = 0);

WR7: SIZEOF(QUERY(pd <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  SIZEOF(QUERY(sa <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    SIZEOF(QUERY(sar <* USEDIN(sa,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT' IN TYPEOF(sar.relateing_shape_aspect))
      AND (sar\shape_aspect_relationship.name = 'stratum feature implementation')
      AND (sar.relateing_shape_aspect.description = 'stratum feature template component')
      )) >= 1
      )) >= 1
      )) >= 1;
END_ENTITY; -- printed_component
(*

```

Formal propositions:

WR1: The printed_component shall have a frame_of_reference with a name of 'layout occurrence'.

WR2: The printed_component shall have a description that is 'printed component'.

WR3: The printed_component shall have exactly one shape_aspect with a name of 'part template occurrence' that is related to a part_template_definition with a description of 'printed part template', 'printed part cross section template', or 'printed connector template'.

WR4: The printed_component shall have at least one shape_aspect that is a component_terminal with a description of 'printed component join terminal'.

WR5: If the printed_component has a description of 'printed connector component', it shall have at least one shape_aspect that is a laminate_component_interface_terminal with a description of 'printed connector component interface terminal'.

WR6: If the printed_component has a description of 'printed connector component', it shall have exactly one shape_aspect with a name of 'part template occurrence' that is related to a part_template_definition with a description of 'printed connector template'.

WR7: The printed_component shall be associated to at least one component_shape_aspect with a description of 'stratum feature template component' as a 'stratum feature implementation'.

5.2.3.1.291 printed_component_link

A printed_component_link is a product_definition and a product_definition_relationship that implements the ARM concept of Printed_component_link (see 4.2.723)

EXPRESS specification:

```

*)
ENTITY printed_component_link
  SUBTYPE OF (product_definition, product_definition_relationship);
WHERE

  WR1: SELF\product_definition_relationship.related_product_definition :<>:
        SELF\product_definition_relationship.relateing_product_definition;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.related_product_definition);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF\product_definition_relationship.relateing_product_definition);

```

```

WR4: sizeof(TYPEOF(SELF) - TYPEOF(SELF\product_definition ||
    SELF\product_definition_relationship ||
    SELF\printed_component_link)) = 0;

WR5: EXISTS(SELF\product_definition.name);

WR6: SELF\product_definition.name = '';

WR7: SELF\product_definition_relationship.name = '';

WR8: sizeof(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')) = 1;
END_ENTITY; -- printed_component_link
(*)

```

Formal propositions:

WR1: The related element shall not be the relating element.

WR2: The printed_component_link shall relate to only a component_definition in the related_product_definition.

WR3: The printed_component_link shall relate to only a component_definition in the relating_product_definition.

WR4: The types of printed_component_link shall be only printed_component_link, product_definition, and product_definition_relationship.

WR5: The name attribute of the product_definition supertype shall exist.

WR6: The name attribute of the product_definition supertype shall be an empty string.

WR7: The name attribute of the product_definition_relationship supertype shall be an empty string.

WR8: Each printed_component_link shall be referenced exactly once by a product_definition_relationship related_product_definition.

5.2.3.1.292 printed_connector_template_terminal_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A printed_connector_template_terminal_relationship is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Printed_connector_template_terminal_relationship (see 4.2.727).

EXPRESS specification:

```

*)
ENTITY printed_connector_template_terminal_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
UNIQUE
  UR1: SELF\shape_aspect_relationship.relate_shape_aspect,
        SELF\shape_aspect_relationship.related_shape_aspect;
WHERE

  WR1: SIZEOF ( QUERY( sar <* USEDIN ( SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (( sar\shape_aspect_relationship.name = 'connector') AND
      (sar.relate_shape_aspect\shape_aspect.description =
        'printed connector template'))
    )) = 1;

  WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL' IN TYPEOF
    (SELF\shape_aspect_relationship.relate_shape_aspect)) AND
    (SELF\shape_aspect_relationship.relate_shape_aspect\shape_aspect.description
    = 'interface terminal'));

  WR3: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL' IN
    TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect)) AND
    (SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description
    = 'join terminal'));

  WR4: SELF\shape_aspect_relationship.relate_shape_aspect :<=:
        SELF\shape_aspect_relationship.related_shape_aspect;
END_ENTITY; -- printed_connector_template_terminal_relationship
(

```

Formal propositions:

UR1: The combination of related and relating shape_aspects shall be unique within a population of printed_connector_template_terminal_relationship.

WR1: The printed_connector_template_terminal_relationship shall be related to a part_template_definition with a description of 'printed connector template'.

WR2: The printed_connector_template_terminal_relationship shall have a relating_shape_aspect that is a printed_part_template_terminal with a description of 'interface terminal'.

WR3: The printed_connector_template_terminal_relationship shall have a related_shape_aspect that is a printed_part_template_terminal with a description of 'join terminal'.

WR4: The related and relating shape aspects shall not be the same shape_aspect.

5.2.3.1.293 printed_part_cross_section_template_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A printed_part_cross_section_template_terminal is a shape_aspect that implements the ARM concept of Printed_part_cross_section_template_terminal (see 4.2.729).

EXPRESS specification:

```

*)
ENTITY printed_part_cross_section_template_terminal
  SUBTYPE OF (printed_part_template_terminal);
WHERE

  WR1:
    SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'cross section definition') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION' IN
      TYPEOF (am.relying_shape_aspect)) AND
      (am.relying_shape_aspect\shape_aspect.description =
      'printed part cross section template')))) = 1;

  WR2: SIZEOF (QUERY (ga <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_TERMINAL_CONNECTION_ZONE_CATEGORY' IN
      TYPEOF (ga.assigned_group)) AND
      (ga.assigned_group.name IN
        ['area edge segment', 'curve edge segment'])))
    )) = 1;

  WR3: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'material to left of terminal') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
      TYPEOF (am.relying_shape_aspect))
    )) <= 1;

```



```

WR4: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'material to right of terminal') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
    TYPEOF (am.relatng_shape_aspect))
    )) <= 1;

WR5: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'material to top of terminal') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
    TYPEOF (am.relatng_shape_aspect))
    )) <= 1;

WR6: SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'material to bottom of terminal') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
    TYPEOF (am.relatng_shape_aspect))
    )) <= 1;

```

```

WR7: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to left of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to right of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to right of terminal') OR
(sar\shape_aspect_relationship.name = 'material to left of terminal')
) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 2);

```

```

WR8: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to left of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to top of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to top of terminal') OR
(sar\shape_aspect_relationship.name = 'material to left of terminal')
) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 2);

```

```

WR9: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to left of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to bottom of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to bottom of terminal') OR
(sar\shape_aspect_relationship.name = 'material to left of terminal')
) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 2);

```

```

WR10: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to top of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to right of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to right of terminal') OR
(sar\shape_aspect_relationship.name = 'material to top of terminal')
) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relatng_shape_aspect))
)) = 2);

```

```

WR11: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to bottom of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to right of terminal') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to right of terminal') OR
(sar\shape_aspect_relationship.name = 'material to bottom of terminal')
) |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relateing_shape_aspect))
)) = 2);

```

```

WR12: NOT((SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to top of terminal') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 1) AND
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'material to bottom of terminal') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 1))
OR
(SIZEOF (QUERY (am <* QUERY (sar <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(sar\shape_aspect_relationship.name = 'material to top of terminal') OR
(sar\shape_aspect_relationship.name = 'material to bottom of terminal')
) |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_TEMPLATE_MATERIAL_LINK' IN
TYPEOF (am.relate_shape_aspect))
)) = 2);
END_ENTITY; -- printed_part_cross_section_template_terminal
( *

```

Formal propositions:

WR1: The `printed_part_cross_section_template_terminal` shall be related to exactly one `part_template_`-definition with a description of 'printed part cross section template' as the 'cross section definition'.

WR2: The `printed_part_cross_section_template_terminal` shall be in exactly one `printed_part_template_`-`terminal_connection_zone_category` that has a name of 'area edge segment' or 'curve edge segment'.

WR3: The `printed_part_cross_section_template_terminal` shall be in no more than one `shape_aspect_`-relationship with a name of 'material to left of terminal' that specifies a `printed_part_template_material_`-link.

WR4: The `printed_part_cross_section_template_terminal` shall be in no more than one `shape_aspect_`-relationship with a name of 'material to right of terminal' that specifies a `printed_part_template_mate-`
`rial_link`.

WR5: The `printed_part_cross_section_template_terminal` shall be in no more than one `shape_aspect_`-relationship with a name of 'material to top of terminal' that specifies a `printed_part_template_material_`-link.

WR6: The `printed_part_cross_section_template_terminal` shall be in no more than one `shape_aspect_`-relationship with a name of 'material to bottom of terminal' that specifies a `printed_part_template_`-`material_link`.

WR7: If the material link to the left of the terminal exists and the material link to the right of the terminal exists, then they shall not be the same material link.

WR8: If the material link to the left of the terminal exists and the material link to the top of the terminal exists, then they shall not be the same material link.

WR9: If the material link to the left of the terminal exists and the material link to the bottom of the terminal exists, then they shall not be the same material link.

WR10: If the material link to the top of the terminal exists and the material link to the right of the terminal exists, then they shall not be the same material link.

WR11: If the material link to the bottom of the terminal exists and the material link to the right of the terminal exists, then they shall not be the same material link.

WR12: If the material link to the top of the terminal exists and the material link to the bottom of the terminal exists, then they shall not be the same material link.

5.2.3.1.294 `printed_part_template_connected_terminals_definition`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `printed_part_template_connected_terminals_definition` is a `shape_aspect` that implements the ARM concept of `Printed_part_template_connected_terminals_definition` (see 4.2.731).

EXPRESS specification:

```
*)
ENTITY printed_part_template_connected_terminals_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1: SELF\shape_aspect.name;
  WHERE

    WR1: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      sar\shape_aspect_relationship.name = 'member connected terminal') |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_TERMINAL' IN
      TYPEOF (mct.related_shape_aspect)) AND
      (mct.related_shape_aspect\shape_aspect.description IN
      ['interface terminal', 'join terminal']))) >= 2;
END_ENTITY; -- printed_part_template_connected_terminals_definition
(*
```


Formal propositions:

UR1: The name shall be unique.

WR1: The `printed_part_template_connected_terminals_definition` shall relate to at least two `printed_part_template_terminals` with a description of 'interface terminal' or 'join terminal' to specify a 'member connected terminal'.

5.2.3.1.295 `printed_part_template_link`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `printed_part_template_link` is a `stratum_technology_link` and `shape_aspect_relationship` that implements the ARM concept of `Printed_part_template_link` (see 4.2.734)

EXPRESS specification:

```

*)
ENTITY printed_part_template_link
  SUBTYPE OF (stratum_technology_link, shape_aspect_relationship);
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

  WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\property_definition_relationship ||
        SELF\printed_part_template_link || SELF\property_definition ||
        SELF\stratum_technology_link || SELF\shape_aspect_relationship)) = 0;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION' IN
        TYPEOF(SELF\shape_aspect_relationship.related_shape_aspect);

  WR4: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION' IN
        TYPEOF(SELF\shape_aspect_relationship.relying_shape_aspect);

  WR5: SELF\shape_aspect_relationship.name = '';

  WR6: SELF\shape_aspect_relationship.description = '';

  WR7: SELF\shape_aspect_relationship.related_shape_aspect.description =
        'printed part template';

  WR8: SELF\shape_aspect_relationship.relying_shape_aspect.description =
        'printed part template';
END_ENTITY; -- printed_part_template_link
(

```

Formal propositions:

WR1: The printed_part_template_link shall not relate to the same shape_aspect through the related and relating attributes.

WR2: The printed_part_template_link shall not be any other subtype of shape_aspect_relationship or property_definition or property_definition_relationship.

WR3: The printed_part_template_link shall reference only a part_template_definition through the related attribute inherited from shape_aspect_relationship.

WR4: The printed_part_template_link shall reference only a part_template_definition through the relating attribute inherited from shape_aspect_relationship.

WR5: The printed_part_template_link name inherited from shape_aspect_relationship shall be blank.

WR6: The printed_part_template_link description inherited from shape_aspect_relationship shall be blank.

WR7: The printed_part_template_link shall reference only a shape_aspect with a description of 'printed part template' through the related attribute inherited from shape_aspect_relationship.

WR8: The printed_part_template_link shall reference only a shape_aspect with a description of 'printed part template' through the relating attribute inherited from shape_aspect_relationship.

5.2.3.1.296 printed_part_template_material

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A printed_part_template_material is a product_definition that implements the ARM concept of: Printed_part_template_material (see 4.2.735)

EXPRESS specification:

*)

```
ENTITY printed_part_template_material
  SUBTYPE OF (product_definition);
WHERE
```

```
WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
  pdr\property_definition_relationship.name = 'technology usage') |
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'STRATUM_TECHNOLOGY' IN
  TYPEOF (tu.relying_property_definition.definition))) = 1)) = 1;
```

```

WR2: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION' IN
    TYPEOF (pdr.used_representation)) |
    ((sr_pdr.used_representation\representation.name =
        'physical characteristics representation')
    AND
    (sizeof (sr_pdr.used_representation.items) = 2 )
    AND
    ( sizeof(QUERY (it <* sr_pdr.used_representation.items |
    it\representation_item.name IN ['maximum width',
        'minimum width']))) = 2 ))
    )) = 1)))
    = 0;

```

```

WR3: sizeof(TYPEOF(SELF) - TYPEOF(SELF\product_definition ||
    SELF\printed_part_template_material)) = 0;

```

```

WR4: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
    TYPEOF (sa)) AND (sa.description = 'top')))) <= 1))) = 0;

```

```

WR5: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
    TYPEOF (sa)) AND (sa.description = 'bottom')))) <= 1))) = 0;

```

```

WR6: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
    TYPEOF (sa)) AND (sa.description = 'left')))) <= 1))) = 0;

WR7: sizeof (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (sa <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
    TYPEOF (sa)) AND (sa.description = 'right')))) <= 1))) = 0;
END_ENTITY; -- printed_part_template_material

(*

```

Formal propositions:

WR1: The printed_part_template_material shall have exactly one property_definition that is related to a property_definition of a stratum_technology as the 'technology usage'.

WR2: The printed_part_template_material shall have exactly one representation that has a name of 'physical characteristics representation' and that has two representation items that have names of 'maximum width', 'minimum width'.

WR3: The printed_part_template_material shall not participate in a complex instance.

WR4: The printed_part_template_material shall not have more than one top boundary.

WR5: The printed_part_template_material shall not have more than one bottom boundary.

WR6: The printed_part_template_material shall not have more than one left boundary.

WR7: The printed_part_template_material shall not have more than one right boundary.

5.2.3.1.297 printed_part_template_material_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A printed_part_template_material_link is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Printed_part_template_material_link (see 4.2.736)

EXPRESS specification:

```

*)
ENTITY printed_part_template_material_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  UNIQUE
    UR1: SELF\shape_aspect_relationship.related_shape_aspect,
        SELF\shape_aspect_relationship.relying_shape_aspect;
  WHERE

    WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect;

    WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

    WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY' IN
        TYPEOF (SELF\shape_aspect_relationship.relying_shape_aspect);

    WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\printed_part_template_material_link)) = 0;

    WR5: SELF\shape_aspect.name = '';

    WR6: SELF\shape_aspect_relationship.name = '';

    WR7: SELF\shape_aspect_relationship.related_shape_aspect.of_shape :<>:
        SELF\shape_aspect_relationship.relying_shape_aspect.of_shape;

    WR8: SELF\shape_aspect_relationship.relying_shape_aspect.description IN
        ['bottom', 'right'];

    WR9: SELF\shape_aspect_relationship.related_shape_aspect.description IN
        ['top', 'left'];
END_ENTITY; -- printed_part_template_material_link
(

```

Formal propositions:

UR1: The relating_shape_aspect, related_shape_aspect combination shall be unique.

WR1: The related boundary shall not be the relating boundary.

WR2: The `printed_part_template_material_link` shall relate to only a `template_material_cross_section_-boundary` in the `related_shape_aspect`.

WR3: The `printed_part_template_material_link` shall relate to only a `template_material_cross_section_-boundary` in the `relating_shape_aspect`.

WR4: The types of `printed_part_template_material_link` shall be only `printed_part_template_material_-link`, `shape_aspect`, and `shape_aspect_relationship`.

WR5: The name attribute of the `shape_aspect` supertype shall be an empty string.

WR6: The name attribute of the `shape_aspect_relationship` supertype shall be an empty string.

WR7: The shape specified by the `related_shape_aspect` shall not be the shape specified by the `relating_shape_aspect`.

WR8: The description attribute of the relating boundary shall be 'bottom' or 'right'.

WR9: The description attribute of the related boundary shall be 'top' or 'left'.

5.2.3.1.298 `printed_part_template_terminal`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `printed_part_template_terminal` is a `shape_aspect` that implements the ARM concepts of:

- `Printed_part_cross_section_template_terminal` (see 4.2.729);
- `Printed_part_template_interface_terminal` (see 4.2.732);
- `Printed_part_template_join_terminal` (see 4.2.733);
- `Printed_part_template_terminal` (see 4.2.737).

EXPRESS specification:

```
*)
ENTITY printed_part_template_terminal
    SUPERTYPE OF (printed_part_cross_section_template_terminal)
    SUBTYPE OF (shape_aspect);
WHERE
```

WR1: SELF\shape_aspect.description IN ['interface terminal', 'join terminal'];

```

WR2: SIZEOF (QUERY (ga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL_CONNECTION_ZONE_CATEGORY' IN
    TYPEOF (ga.assigned_group))) = 1;

WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    sar.related_shape_aspect\shape_aspect.description =
    'connection zone')) >= 1;

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' IN
    TYPEOF (pdr.used_representation))) >= 1))) = 0;

WR5: SIZEOF (QUERY (ad <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated definition') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN
    TYPEOF (ad.relying_shape_aspect)) AND
    ((ad.relying_shape_aspect\shape_aspect.description = 'printed part template')
    OR
    (ad.relying_shape_aspect\shape_aspect.description = 'printed connector template'))
    )) = 1;

WR6: SIZEOF (QUERY (mct <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'member connected terminal') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION' IN
    TYPEOF (mct.relying_shape_aspect))) <= 1;
END_ENTITY; -- printed_part_template_terminal
(*

```

Formal propositions:

WR1: The printed_part_template_terminal shall have a description of 'interface terminal' or 'join terminal'.

WR2: The `printed_part_template_terminal` shall be in exactly one `printed_part_template_terminal_connection_zone_category`.

WR3: The `printed_part_template_terminal` shall relate to at least one 'connection zone'.

WR4: The `printed_part_template_terminal` shall have at least one property that is represented by a `shape_representation`.

WR5: The `printed_part_template_terminal` shall be related to exactly one `part_template_definition` with a description of 'printed part template' or 'printed connector template' as the 'associated definition'.

WR6: The `printed_part_template_terminal` shall be related to at most one `printed_part_template_connected_terminals_definition` as the 'member connected terminal'.

NOTE Inverse cardinality rules required by Application Objects are implemented in rule `printed_connector_template_terminal_relationship_unique_constraint`.

5.2.3.1.299 `printed_part_template_terminal_connection_zone_category`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `printed_part_template_terminal_connection_zone_category` is a group that specifies either an `area_edge_segment`, a `curve_edge_segment`, a `surface_area`, or a `surface_point` for a `Printed_part_template_terminal` (see 4.2.737).

EXPRESS specification:

```
*)
ENTITY printed_part_template_terminal_connection_zone_category
    SUBTYPE OF (group);
WHERE

    WR1: SELF\group.name IN ['area edge segment', 'curve edge segment',
        'surface area', 'surface point'];
END_ENTITY; -- printed_part_template_terminal_connection_zone_category
(*)
```

Formal propositions:

WR1: The `printed_part_template_terminal_connection_zone_category` shall have a name equal to 'area edge segment', 'curve edge segment', 'surface area', or 'surface point'.

5.2.3.1.300 `probe_access_area`

A `probe_access_area` is a `component_shape_aspect` that implements the ARM concepts of `Internal_probe_access_area` (see 4.2.475) and `Probe_access_area` (see 4.2.739).

EXPRESS specification:

```

*)
ENTITY probe_access_area
  SUBTYPE OF (component_shape_aspect);
WHERE

  WR1: EXISTS(SELF.of_shape.definition\product_definition.name);

  WR2: ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition)) AND
        (SELF.of_shape.definition\product_definition.name = 'interconnect module');

  WR3: SIZEOF (QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        sar.related_shape_aspect\shape_aspect.description =
        'connection zone')) = 1;

  WR4: SIZEOF (QUERY (pli <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'probed layout item') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE' IN
        TYPEOF (pli.related_shape_aspect))) = 1;

  WR5: (NOT (SELF\shape_aspect.description =
        'internal probe access area')) OR
        (SIZEOF (QUERY (i <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'implementation') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE' IN
        TYPEOF (i.relateing_shape_aspect)))) = 1);
END_ENTITY; -- probe_access_area
(

```

Formal propositions:

WR1: The name of the product_definition associated with this probe_access_area shall exist.

WR2: The probe_access_area shall be an aspect of the shape of a component_definition that is a definition with a name of 'interconnect module'.

WR3: The probe_access_area shall relate to exactly one shape_aspect with a description of 'connection zone'.

WR4: The probe_access_area shall relate to exactly one stratum_feature to specify a 'probed layout item'.

WR5: If the probe_access_area has a description of 'internal probe access area', the probe_access_area shall be related to exactly one stratum_feature as the 'implementation'.

5.2.3.1.301 product_related_characterized_product_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A product_related_characterized_product_category is a characterized_product_category and a product_related_product_category that implements the ARM concept of Ee_product_related_characterized_product_category (see ??) .

EXPRESS specification:

```
*)
ENTITY product_related_characterized_product_category
    SUBTYPE OF (characterized_product_category,
                product_related_product_category);
END_ENTITY; -- product_related_characterized_product_category
(*
```

5.2.3.1.302 product_specific_parameter_value_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A product_specific_parameter_value_assignment is a characterized_product_category and a product_related_product_category that implements the ARM concept of Ee_product_specific_parameter_value_assignment (see 4.2.327) .

EXPRESS specification:

```
*)
ENTITY product_specific_parameter_value_assignment
    SUBTYPE OF (product_related_characterized_product_category);
WHERE
```

WR1: sizeof(SELF\product_related_product_category.products) = 1;

```

WR2: sizeof(query(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (pd\property_definition.description =
        'product category parameter') AND
    (sizeof(query(pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (sizeof (
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODEL_PARAMETER',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'] *
    TYPEOF (pdr.used_representation)) = 1))) = 1))) = 1;

WR3: sizeof(query(pcr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY') |
    ((pcr\product_category_relationship.description =
        'product value assignment') AND
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY') IN
    TYPEOF (pcr.category))
    ))) = 1;
END_ENTITY; -- product_specific_parameter_value_assignment
(*

```

Formal propositions:

WR1: There shall be exactly one product in the role of products.

WR2: It shall be the definition of exactly one property_definition with a description of 'product category parameter' that is the definition of a property_definition_representation with a used_representation that is a representation that has in its items exactly one model_parameter, exactly one measure_representation_item, or exactly one descriptive_representation_item.

WR3: It shall be the sub_category of exactly one category that is a product_related_product_category, and the product_category_relationship establishing the sub_category relationship shall have a description of 'product value assignment'.

5.2.3.1.303 promissory_usage_in_product_model

A promissory_usage_in_product_model is a group that participates in an assignment to indicate that a product_definition is to be associated with a product_model. The promissory_usage_in_product_model string attributes may be populated with 'undefined' to indicate there is no requirement associated with these attributes.

NOTE The group entity is provided for structural consistency with other parts of this standard.

EXPRESS specification:

```

*)
ENTITY promissory_usage_in_product_model
  SUBTYPE OF (group);
WHERE

  WR1: SIZEOF( QUERY(aga <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PROMISSORY_USAGE_IN_PRODUCT_MODEL_ASSIGNMENT' IN TYPEOF(aga))
    ) >= 1;
END_ENTITY; -- promissory_usage_in_product_model
( *
```

Formal propositions:

WR1: The `promissory_usage_in_product_model` shall be referenced by an `applied_promissory_usage_in_product_model_assignment`.

5.2.3.1.304 rectangular_array_placement_group_component_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `rectangular_array_placement_group_component_definition` is a `array_placement_group_component_definition` that implements the ARM concept of `Rectangular_array_placement_group` (see ??) .

EXPRESS specification:

```

*)
ENTITY rectangular_array_placement_group_component_definition
  SUBTYPE OF (array_placement_group_component_definition);
END_ENTITY; -- rectangular_array_placement_group_component_definition
( *
```

5.2.3.1.305 rectangular_array_placement_group_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An `rectangular_array_placement_group_component_shape_aspect` is a `array_placement_group_component_shape_aspect` that implements the ARM concepts of `Rectangular_array_placement_group` (see ??)

EXPRESS specification:

```
*)
ENTITY rectangular_array_placement_group_component_shape_aspect
  SUBTYPE OF (array_placement_group_component_shape_aspect);
END_ENTITY; -- rectangular_array_placement_group_component_shape_aspect
(*
```

5.2.3.1.306 `rectangular_composite_array_shape_aspect`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `rectangular_composite_array_shape_aspect` is a `composite_shape_aspect` that implements the ARM concept of `Part_rectangular_array_feature` (see 4.2.619).

EXPRESS specification:

```
*)
ENTITY rectangular_composite_array_shape_aspect
  SUBTYPE OF (composite_array_shape_aspect);
END_ENTITY; -- rectangular_composite_array_shape_aspect
(*
```

5.2.3.1.307 `reference_packaged_part_assembly_implementation`

A `reference_packaged_part_assembly_implementation` is a `physical_unit`. A `reference_packaged_part_assembly_implementation` implements the ARM concept of a `Reference_packaged_part_assembly_implementation` (see 4.2.766).

EXPRESS specification:

```
*)
ENTITY reference_packaged_part_assembly_implementation
  SUBTYPE OF (physical_unit);
WHERE
```

```

WR1: (SIZEOF (QUERY (prpc <*
    USEDIN (SELF.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
    prpc\product_category.name = 'template model')) >= 1) AND
    (SELF\product_definition.name = 'assembly module'));

WR2: (NOT (SELF.frame_of_reference.name = 'physical design')) OR
    (SIZEOF (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' +
    'RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'design usage') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_ASSEMBLY_DEFINITION'] *
    TYPEOF (du.relying_product_definition)) = 1) AND
    (du.relying_product_definition.frame_of_reference.name =
    'physical design usage') AND
    (du.relying_product_definition\product_definition.name =
    'assembly module') AND
    (SIZEOF (QUERY (prpc <*
    USEDIN (du.relying_product_definition.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
    prpc\product_category.name = 'template model')) >= 1)
    )) = 1);
END_ENTITY; -- reference_packaged_part_assembly_implementation
(*

```

Formal propositions:

WR1: The reference_packaged_part_assembly_implementation name shall be 'assembly model' and it shall be a definition of a product that is classified as an 'template model'.

WR2: If the reference_packaged_part_assembly_implementation is a physical design, it shall relate to exactly one reference_packaged_part_assembly_implementation, externally_defined_reference_packaged_part_assembly_implementation, or library_defined_reference_packaged_part_assembly_implementation that is a physical design usage with a frame_of_reference name of 'assembly module' as the design usage.

5.2.3.1.308 reference_packaged_part_interconnect_implementation

A reference_packaged_part_interconnect_implementation is a physical_unit that implements the ARM concept of Reference_packaged_part_interconnect_implementation (see 4.2.767).

EXPRESS specification:

```

*)
ENTITY reference_packaged_part_interconnect_implementation
  SUBTYPE OF (physical_unit);
WHERE

  WR1: EXISTS(SELF\product_definition.name);

  WR2: (SIZEOF (QUERY (prpc <*
    USEDIN (SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
    prpc\product_category.name = 'template model')) >= 1) AND
    (SELF\product_definition.name = 'interconnect module');

  WR3: SIZEOF (QUERY (du <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' +
    'RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'design usage') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PHYSICAL_UNIT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PHYSICAL_UNIT'] *
    TYPEOF (du.relying_product_definition)) = 1) AND
    (du.relying_product_definition.frame_of_reference.name =
      'physical design usage') AND
    (du.relying_product_definition\product_definition.name =
      'interconnect module') AND
    (SIZEOF (QUERY (prpc <*
      USEDIN (du.relying_product_definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
      prpc\product_category.name = 'template model')) >= 1)
    )) <= 1;

  WR4: SELF.frame_of_reference.name = 'physical design';
END_ENTITY; -- reference_packaged_part_interconnect_implementation
(

```

Formal propositions:

WR1: The name attribute shall exist.

WR2: The reference_packaged_part_interconnect_implementation name shall be 'interconnect module' and it shall be a definition of a product that is classified as an 'template model'.

WR3: The reference_packaged_part_interconnect_implementation shall relate to no more than one physical_unit, externally_defined_physical_unit, or library_defined_physical_unit that is a physical design usage with a name of 'interconnect module' as the design usage, and that is a 'template model'.

WR4: The reference_packaged_part_interconnect_implementation shall be in the 'physical design' context.

5.2.3.1.309 requirement_allocation_group

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A requirement_allocation_group is a group and a property_definition_relationship that implements the ARM concept of Requirement_allocation (see 4.2.769) and Test_requirement_allocation (see 4.2.895).

EXPRESS specification:

```

*)
ENTITY requirement_allocation_group
  SUBTYPE OF (group, property_definition_relationship);
WHERE

  WR1: (SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT' IN
    TYPEOF (ga)) |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_FORMATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONFIGURATION_ITEM'] * TYPEOF (aga.items)) = 1))) = 1);

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN TYPEOF
    (SELF\property_definition_relationship.relying_property_definition);

```



```

WR4: NOT(SELf\property_definition_relationship.related_property_definition.description =
  (SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT' IN
    TYPEOF (ga)) |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION'] * TYPEOF (aga.items)) = 1))) >= 1);
END_ENTITY; -- requirement_allocation_group
(*

```

Formal propositions:

WR1: The requirement_allocation_group shall be assigned to exactly one item that is a configuration_item, product, or product_definition_formation.

WR2: The requirement_allocation_group shall reference a requirements_property as the relating_property_definition.

WR3: If the property_definition_relationship references a property_definition with a description of 'test requirement' the requirement_allocation_group shall be assigned to more than one property_definition_representation.

5.2.3.1.310 requirement_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

An requirement_definition is a product_definition that implements the ARM concepts of: Requirement_definition (see 4.2.771)

EXPRESS specification:

```

*)
ENTITY requirement_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: (NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF (SELF))) OR
    (SIZEOF (QUERY (docs <*
    SELf\product_definition_with_associated_documents.
    documentation_ids |
    docs.kind\document_type.product_data_type = 'CAD filename')) <= 1);

```

```

WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    adta.role\date_time_role.name = 'creation date')) +
    SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ada.role\date_role.name = 'creation date')) = 1;

WR3: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1;

WR4: SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\person_and_organization_role.name = 'creator')) +
    SIZEOF (QUERY (apoa <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    apoa.role\organization_role.name = 'creator')) >= 1;

WR5: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1;

WR6: SIZEOF (TYPEOF (SELF) - (TYPEOF (SELF\product_definition ||
    SELF\requirement_definition))) = 0;

WR7: SIZEOF (QUERY (prpc <* USEDIN (SELF.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
    'PRODUCTS') |
    prpc\product_category.name = 'requirements model')) >= 1;
END_ENTITY; -- requirement_definition
(*)

```

Formal propositions:

WR1: If the requirement_definition is a product_definition_with_associated_documents, there shall be at most one document in the set of documentation_ids that has a product_data_type of 'CAD filename'

WR2: The requirement_definition shall have exactly one creation date.

WR3: The requirement_definition shall have exactly one approval.

WR4: The requirement_definition shall have at least one organization or person_and_organization as the creator.

WR5: The requirement_definition shall have exactly one security_classification.

WR6: The requirement_definition shall not be any other type of product_definition.

WR7: The requirement_definition shall have a product_category with a name of 'requirements model'.

5.2.3.1.311 requirements_property

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A requirements_property is a property_definition that implements the ARM concepts of:

- Constraint_occurrence (see 4.2.193);
- Design_characteristic_occurrence (see 4.2.261);
- Design_requirement_occurrence (see 4.2.279);
- Ee_requirement_occurrence (see 4.2.330);
- Electrical_requirement_occurrence (see 4.2.346);
- Electromagnetic_requirement_occurrence (see 4.2.348);
- Interface_requirement_occurrence (see 4.2.471);
- Material_electrical_conductivity_requirement_occurrence (see 4.2.532);
- Part_based_constraint_occurrence (see ??);
- Purpose_occurrence (see 4.2.757);
- Thermal_requirement_occurrence (see 4.2.908).

EXPRESS specification:

*)

```
ENTITY requirements_property
  SUBTYPE OF (property_definition);
WHERE
```

```
  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'requirements property composition')) <
```

```

WR2: SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF (QUERY (it <* pdr.used_representation.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND
    (it\representation_item.name = 'requirements description')))) = 1)) <= 1;

WR3: SIZEOF (QUERY (dr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION' IN
    TYPEOF (dr.assigned_document))) >= 1;

WR4: NOT(EXISTS(SELF\property_definition.description)) OR
    (NOT (SELF\property_definition.description IN ['constraint',
    'part based constraint']) OR
    (SIZEOF (QUERY (dc <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'design constraint') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_DEFINITION' IN
    TYPEOF (dc.relatng_property_definition.definition))) = 1));

WR5: NOT(EXISTS(SELF\property_definition.description)) OR
    (NOT (SELF\property_definition.description =
    'part based constraint') OR
    (SIZEOF (QUERY (cp <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'constraining part') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP' IN
    TYPEOF (cp.relatng_property_definition.definition)) AND
    (cp.relatng_property_definition.definition\product_definition_relationship.name =
    'constraining part') AND
    (cp.relatng_property_definition.definition.related_product_definition.
    frame_of_reference.name = 'design requirement')))) = 1));

```

```

WR6: NOT(EXISTS(SELF\property_definition.description)) OR
      (NOT (SELF\property_definition.description =
        'interface requirement') OR
        (SIZEOF (QUERY (itnha <* QUERY (pdr <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
        pdr\property_definition_relationship.name = 'interface to next higher assembly') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION' IN
        TYPEOF (itnha.relying_property_definition.definition)) AND
        (itnha.relying_property_definition.definition.frame_of_reference.name
        = 'design requirement') AND
        (SIZEOF (QUERY (hai <* QUERY (pdr <*
        USEDIN (itnha.relying_property_definition.definition,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        pdr\product_definition_relationship.name = 'higher assembly interface') |
        SIZEOF (QUERY (pdr <* USEDIN (hai,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SPECIFIED_HIGHER_USAGE_OCCURRENCE'
        IN TYPEOF (pdr))) = 1)) = 1))) = 1));

WR7: SIZEOF (QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      SIZEOF (QUERY (it <* pdr.used_representation.items |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND
      (it\representation_item.name = 'requirements name')))) = 1)) = 1;
END_ENTITY; -- requirements_property
(*)

```

Formal propositions:

WR1: The requirements_property shall be related to at most one property_definition as the 'requirements property composition'.

WR2: The requirements_property shall have at most one representation that contains a descriptive_representation_item with a name of 'requirements description'.

WR3: The requirements_property shall be referenced by at least one ee_specification.

WR4: If the requirements_property has a description of 'constraint' or 'part based constraint', it shall be related to exactly one property_definition of a rule_definition as the 'design constraint'.

WR5: If the requirements_property has a description of 'part based constraint', it shall be related to exactly one property_definition of a product_definition_relationship with name of 'constraining part' and a related_product_definition that has a frame_of_reference name of 'design requirement' as the 'constraining part'.

WR6: If the requirements_property has a description of 'interface requirement', it shall be related to exactly one property_definition of a product_definition that has a frame_of_reference name of 'design requirement', and that relates exactly one product_definition to a product_definition that is the component in exactly one specified_higher_usage_occurrence to specify the 'higher assembly interface', as the 'interface to next higher assembly'.

WR7: The requirements_property shall have exactly one representation that contains a descriptive_representation_item with a name of 'requirements name'.

5.2.3.1.312 requirements_property_group

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A requirements_property_group is a requirements_property and a group that implements the ARM concept of Requirement_group_occurrence (see 4.2.772).

EXPRESS specification:

```
*)
ENTITY requirements_property_group
  SUBTYPE OF (requirements_property, group);
WHERE

  WR1: SIZEOF (QUERY (rpc <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'requirements property composition') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN
    TYPEOF (rpc.related_property_definition))) >= 1;
END_ENTITY; -- requirements_property_group
(*
```

Formal propositions:

WR1: The requirements_property_group shall relate to at least one requirements_property as a 'requirements property composition'.

5.2.3.1.313 resistance_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `resistance_measure_with_unit` is a `measure_with_unit` that implements the ARM concept of `Resistance_measure_with_unit` (see ??).

EXPRESS specification:

```

*)
ENTITY resistance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\resistance_measure_with_unit))) = 0;

  WR2: SELF\measure_with_unit.unit_component\si_unit.name = ohm;

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.SI_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- resistance_measure_with_unit
(

```

Formal propositions:

WR1: The `resistance_measure_with_unit` shall not be any other type of `measure_with_unit`.

WR2: The `resistance_measure_with_unit` shall reference an `si_unit` with a name of `ohm`.

WR3: The `unit_component` shall be an `si_unit`.

5.2.3.1.314 **routed_physical_component**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `routed_physical_component` is a `component_definition` that implements the ARM concept of `Routed_physical_component` (see 4.2.778)

EXPRESS specification:

```

*)
ENTITY routed_physical_component
  SUBTYPE OF (component_definition);
WHERE

```

```

WR1: SIZEOF (QUERY (ip <* QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
    pdr\product_definition_relationship.name = 'instantiated part') |
    (SIZEOF (['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT'] *
    TYPEOF (ip.relatinq_product_definition)) = 1) AND
    (ip.relatinq_product_definition.frame_of_reference.name =
    'physical design usage')))) = 1;

WR2: SELF\product_definition.description :<>: 'laminate component';
END_ENTITY; -- routed_physical_component
(*)

```

Formal propositions:

WR1: The routed_physical_component shall be related, as the 'instantiated part', to exactly one physical_unit that is in the 'physical design usage' product_definition_context.

WR2: The routed_physical_component shall not have a description of 'laminate component'.

5.2.3.1.315 routed_printed_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A routed_printed_component is a printed_component that implements the ARM concept of Routed_printed_component (see 4.2.779).

EXPRESS specification:

```

*)
ENTITY routed_printed_component
  SUBTYPE OF (printed_component);
WHERE

```



```

WR1: sizeof(query(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (sizeof(query(sa <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') |
    (sizeof(query(sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN TYPEOF(sar.relate_shape_aspect))
    AND (sar.relate_shape_aspect\shape_aspect.description =
    'printed part cross section template')
    AND (sar\shape_aspect_relationship.name = 'instantiated template')
    )) = 1)
    )) = 1)
    )) = 1;

WR2: sizeof(query(csa <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN TYPEOF(csa))
    AND (sizeof(query(sar <* USEDIN(csa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (sizeof(query(pdr <* USEDIN(sar,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(pdr)) AND
    (pdr\product_definition_relationship.name = 'inter stratum extent')
    )) = 1)
    AND (sar\shape_aspect_relationship.name = 'component extent')
    )) = 1)
    )) = 1;

END_ENTITY; -- routed_printed_component
(*

```

Formal propositions:

WR1: The routed_printed_component shall be associated to exactly one part_template_definition with a description of 'printed part cross section template' as an 'instantiated template'.

WR2: The routed_printed_component shall be associated to exactly one component_shape_aspect that shall be related to exactly one shape_aspect that is a shape_aspect for a product_definition_relationship with a name of 'inter stratum extent' to specify a 'component extent'.

Informal propositions:

IP1: The routed_printed_component shall have exactly one property that has exactly one representation.

5.2.3.1.316 rule_action

A rule_action is an action that implements the ARM concepts of:

- Rule_action (see 4.2.780);
- Rule_adoption (see 4.2.781);
- Rule_change_request (see 4.2.784);
- Rule_creation (see 4.2.785);
- Rule_expiration (see 4.2.786);
- Rule_justification (see 4.2.787);
- Rule_modification (see 4.2.788);
- Rule_rejection (see 4.2.790);
- Rule_request (see 4.2.791);
- Rule_submission (see 4.2.794);
- Rule_supersedence (see 4.2.795).

EXPRESS specification:

```
*)  
ENTITY rule_action  
  SUBTYPE OF (action);  
WHERE
```

```

WR1: SIZEOF (QUERY (aaa <* QUERY (aa <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_ACTION_ASSIGNMENT' IN
TYPEOF (aa)) |
SIZEOF (QUERY (it <* aaa.items |
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_FORMATION' IN
TYPEOF (it))
AND (it\product_definition_formation.description = 'rule version'))
)) = 1)) = 1;

WR2: SIZEOF (QUERY (adta <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
adta.role\date_time_role.name = 'participant date and time')) +
SIZEOF (QUERY (ada <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DATE_ASSIGNMENT.ITEMS') |
ada.role\date_role.name = 'participant date')) = 1;

WR3: (NOT (SELF\action.name = 'rule justification')) OR
(SIZEOF (QUERY (ja <* QUERY (ar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ACTION_RELATIONSHIP.RELATED_ACTION') |
ar\action_relationship.name = 'justified action') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'RULE_ACTION' IN
TYPEOF (ja.relating_action))) = 1);

WR4: (NOT (SELF\action.name = 'rule modification')) OR
(SIZEOF (QUERY (mr <* QUERY (ar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ACTION_RELATIONSHIP.RELATED_ACTION') |
ar\action_relationship.name = 'modification rationale') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'RULE_ACTION' IN
TYPEOF (mr.relating_action)) AND
(mr.relating_action\action.name = 'rule change request')) = 1);

```

```

WR5: (NOT (SELF\action.name = 'rule replacement from')) OR
      (SIZEOF (QUERY (rrfa <* QUERY (aa <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_REPLACEMENT_FROM_ASSIGNMENT' IN
        TYPEOF (aa)) |
        SIZEOF (QUERY (it <* rrfa.items |
          (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_DEFINITION_FORMATION' IN
          TYPEOF (it))
        AND (it\product_definition_formation.description = 'rule version'))
      )) = 1)) = 1);

WR6: (NOT (SELF\action.name = 'rule replacement to')) OR
      (SIZEOF (QUERY (rrta <* QUERY (aa <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_REPLACEMENT_TO_ASSIGNMENT' IN
        TYPEOF (aa)) |
        SIZEOF (QUERY (it <* rrta.items |
          (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_DEFINITION_FORMATION' IN
          TYPEOF (it))
        AND (it\product_definition_formation.description = 'rule version'))
      )) = 1)) = 1);
END_ENTITY; -- rule_action
(*

```

Formal propositions:

WR1: The rule_action shall be assigned to exactly one product_definition_formation with a description of 'rule version'.

WR2: The rule_action shall have exactly one date or one date and time with a role of 'participant date' or 'participant date and time'.

WR3: If the rule_action has a name of 'rule justification', it shall be related to rule_action as the 'justified action'.

WR4: If the rule_action has a name of 'rule modification', it shall be related to rule_action with a name of 'rule change request' as the 'modification rationale'.

WR5: If the rule_action has a name of 'rule replacement from', it shall be assigned to exactly one product_definition_formation with a description of 'rule version' by a rule_replacement_from_assignment.

WR6: If the rule_action has a name of 'rule replacement to', it shall be assigned to exactly one product_definition_formation with a description of 'rule version' by a rule_replacement_to_assignment.

5.2.3.1.317 rule_complex_clause

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A rule_complex_clause is a characterized_object and is an externally_defined_item that implements the ARM concept of Ee_rule_complex_clause (see 4.2.331).

EXPRESS specification:

```

*)
ENTITY rule_complex_clause
  SUBTYPE OF (representation);
  UNIQUE
    UR1: SELF\representation.name;
  WHERE

    WR1: SIZEOF (QUERY (rr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_2') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_SIMPLE_CLAUSE' IN TYPEOF (rr.rep_1))) >= 1;

    WR2: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\representation ||
      SELF\rule_complex_clause)) = 0;
END_ENTITY; -- rule_complex_clause
(

```

Formal propositions:

UR1: The name shall be unique within a population of rule_complex_clause.

WR1: The rule_complex_clause shall have at least one representation_relationship that references a rule_simple_clause.

WR2: The rule_complex_clause shall not be a complex instance.

5.2.3.1.318 rule_conclusion_definition

A rule_conclusion_definition is a representation that implements the ARM concept of Ee_rule_conclusion_definition (see 4.2.332).

EXPRESS specification:

```

*)
ENTITY rule_conclusion_definition
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_DEFINITION' IN TYPEOF
    (pdr.definition.definition))) = 1;

  WR2: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_SIMPLE_CLAUSE' IN TYPEOF
    (rr\representation_relationship.rep_1))) >= 1;

  WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\representation ||
    SELF\rule_conclusion_definition)) = 0;
END_ENTITY; -- rule_conclusion_definition
(*

```

Formal propositions:

WR1: The rule_conclusion_definition shall have one property_definition_representation that references a rule definition.

WR2: The rule_conclusion_definition shall have at least one representation_relationship that references a rule_simple_clause.

WR3: The rule_conclusion_definition shall not be a complex instance.

5.2.3.1.319 rule_definition

A rule_definition is a product_definition that implements the ARM concept of Ee_rule_definition (see 4.2.333).

EXPRESS specification:

```

*)
ENTITY rule_definition
  SUBTYPE OF (product_definition);
WHERE

  WR1: SELF\product_definition.formation.description = 'rule version';
END_ENTITY; -- rule_definition
( *

```

Formal propositions:

WR1: The rule_definition shall reference a product_definition_formation with a description of 'rule version'.

5.2.3.1.320 rule_function_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A rule_function_definition is a representation that implements the ARM concept of Ee_rule_function (see 4.2.334).

EXPRESS specification:

```

*)
ENTITY rule_function_definition
  SUBTYPE OF (representation);
WHERE

  WR1: SIZEOF (QUERY (it <* SELF.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODEL_PARAMETER' IN
    TYPEOF (it))) >= 1;

  WR2: SIZEOF(QUERY(adf <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (adf\document_reference.assigned_document.kind\document_type.product_data_type
    = 'reference document') AND
    (SIZEOF([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DOCUMENT' ] *
    TYPEOF(adf\document_reference.assigned_document))>=1))) = 1;

```

```

WR3: SIZEOF(QUERY(adf <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (adf\document_reference.assigned_document.kind\document_type.product_data_type =
    'source code') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION' IN
    TYPEOF(adf\document_reference.assigned_document)))) = 1;
END_ENTITY; -- rule_function_definition
(*

```

Formal propositions:

WR1: The rule_function_definition shall contain at least one model_parameter.

WR2: The rule_function_definition shall contain exactly one document or ee_specification with a document_type of 'reference document'.

WR3: The rule_function_definition shall contain exactly one ee_specification with a document_type of 'source code'.

Informal propositions:

IP1: The combination of identifier and the item which has a document type of 'reference document' shall be unique for rule_function_definition.

5.2.3.1.321 rule_premise_definition

A rule_premise_definition is a representation that implements the ARM concept of Ee_rule_premise_definition (see 4.2.335).

EXPRESS specification:

```

*)
ENTITY rule_premise_definition
  SUBTYPE OF (representation);
WHERE

WR1: SIZEOF (QUERY (pdr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_DEFINITION' IN TYPEOF
    (pdr.definition.definition))) = 1;

```



```

WR2: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_COMPLEX_CLAUSE' IN TYPEOF
    (rr\representation_relationship.rep_1))) >= 1;

WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\representation ||
    SELF\rule_premise_definition)) = 0;
END_ENTITY; -- rule_premise_definition
(*

```

Formal propositions:

WR1: The rule_premise_definition shall have one property_definition_representation that references a rule definition.

WR2: The rule_premise_definition shall have at least one representation_relationship that references a rule_complex_clause.

WR3: The rule_premise_definition shall not be a complex instance.

5.2.3.1.322 rule_replacement_from_assignment

The rule_replacement_from_assignment is an action_assignment that specifies a rule version which is being superseded in a Rule_supersedence (see 4.2.795).

EXPRESS specification:

```

*)
ENTITY rule_replacement_from_assignment
    SUBTYPE OF (action_assignment);
    items : SET [1:?] OF replacement_from_item;
END_ENTITY; -- rule_replacement_from_assignment
(*

```

5.2.3.1.323 rule_replacement_to_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A rule_replacement_to_assignment is an action_assignment that specifies a rule version which is superseding another rule version in a Rule_supersedence (see 4.2.795).

EXPRESS specification:

```
*)
ENTITY rule_replacement_to_assignment
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF replacement_to_item;
END_ENTITY; -- rule_replacement_to_assignment
(*
```

Informal propositions:

IP1: The assigned_action shall be in the role of a related_action for an action_relationship where the relating_action is an action that has been assigned by a rule_replacement_from_assignment.

5.2.3.1.324 rule_set

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A rule_set is a group that implements the ARM concepts of Rule_set (see 4.2.792) and Rule_set_group (see 4.2.793).

EXPRESS specification:

```
*)
ENTITY rule_set
  SUBTYPE OF (group);
  UNIQUE
    UR1: SELF\group.name;
WHERE
```

```

WR1: (SIZEOF (QUERY (aga <* QUERY (ga <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_GROUP_ASSIGNMENT' IN
TYPEOF (ga)) |
SIZEOF (QUERY (rd <* QUERY (it <* aga.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'RULE_DEFINITION' IN
TYPEOF (it)) |
SIZEOF (QUERY (pd <* USEDIN (rd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'rule priority')) >= 1)) >= 1)) >= 1)) >= 1) OR
(SIZEOF (QUERY (rsge <* QUERY (gr <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_RELATIONSHIP.RELATING_GROUP') |
gr\group_relationship.name = 'rule set group element') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'RULE_SET' IN
TYPEOF (rsge.related_group))) >= 2);
END_ENTITY; -- rule_set
(*

```

Formal propositions:

UR1: The name shall be unique.

WR1: The rule_set shall be assigned to at least one rule_definition that has at least one property that has at least one representation with a name of 'rule priority', or it shall relate to at least one rule_sets to specify a 'rule set group element'.

5.2.3.1.325 rule_simple_clause

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A rule_simple_clause is a representation that implements the ARM concept of Ee_rule_simple_clause (see 4.2.336).

EXPRESS specification:

```

*)
ENTITY rule_simple_clause
  SUBTYPE OF (representation);
  UNIQUE
    UR1: SELF\representation.name;
  WHERE

    WR1: SIZEOF (QUERY (rr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_1') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PARAMETER_ASSIGNMENT_REPRESENTATION' IN
      TYPEOF (rr.rep_2))) >= 1;

    WR2: SIZEOF (QUERY (rr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_2') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_FUNCTION_DEFINITION' IN TYPEOF (rr.rep_1))) = 1;

    WR3: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\representation ||
      SELF\rule_simple_clause)) = 0;
END_ENTITY; -- rule_simple_clause
(*)

```

Formal propositions:

UR1: The name shall be unique within a population of rule_simple_clause.

WR1: The rule_simple_clause shall have at least one representation_relationship that references a parameter_assignment_representation.

WR2: The rule_simple_clause shall have exactly one representation_relationship that references a rule_function_definition.

WR3: The rule_simple_clause shall not be a complex instance.

5.2.3.1.326 scalar_terminal_definition_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A scalar_terminal_definition_link is a shape_aspect and a shape_aspect_relationship that implements the ARM concept of Scalar_terminal_definition_link (see 4.2.800)

EXPRESS specification:

```

*)
ENTITY scalar_terminal_definition_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
UNIQUE
  UR1: related_shape_aspect, relating_shape_aspect;
WHERE

  WR1: SELF\shape_aspect_relationship.related_shape_aspect :<>:
        SELF\shape_aspect_relationship.relatng_shape_aspect;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' IN
        TYPEOF (SELF\shape_aspect_relationship.related_shape_aspect);

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT_TERMINAL_DEFINITION' IN
        TYPEOF (SELF\shape_aspect_relationship.relatng_shape_aspect);

  WR4: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
        SELF\shape_aspect_relationship ||
        SELF\scalar_terminal_definition_link)) = 0;

  WR5: SELF\shape_aspect.name = '';

  WR6: SELF\shape_aspect_relationship.name = '';

  WR7: SELF\shape_aspect_relationship.relatng_shape_aspect\shape_aspect.description = 'sc

  WR8: SELF\shape_aspect_relationship.related_shape_aspect\shape_aspect.description = 'sc
END_ENTITY; -- scalar_terminal_definition_link
(*)

```

Formal propositions:

UR1: The related_shape_aspect in combination with the relating shape_aspect shall be unique.

WR1: The related point shall not be the relating point.

WR2: The scalar_terminal_definition_link shall relate to only a functional_unit_terminal_definition in the related_shape_aspect.

WR3: The scalar_terminal_definition_link shall relate to only a functional_unit_terminal_definition in the relating_shape_aspect.

WR4: The types of scalar_terminal_definition_link shall be only scalar_terminal_definition_link, shape_aspect, and shape_aspect_relationship.

WR5: The name attribute of the shape_aspect supertype shall be an empty string.

WR6: The name attribute of the shape_aspect_relationship supertype shall be an empty string.

WR7: The scalar_terminal_definition_link shall relate to only a functional_unit_terminal_definition in the relating_shape_aspect that has a description of 'scalar terminal'.

WR8: The scalar_terminal_definition_link shall relate to only a functional_unit_terminal_definition in the related_shape_aspect that has a description of 'scalar terminal'.

5.2.3.1.327 scattering_parameter_measure_with_unit

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A scattering_parameter_measure_with_unit is a measure_with_unit that implements the ARM concept of Scattering_parameter_measure_with_unit (see ??).

EXPRESS specification:

```
*)
ENTITY scattering_parameter_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\measure_with_unit ||
    SELF\scattering_parameter_measure_with_unit))) = 0;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.RATIO_UNIT' IN
    TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY; -- scattering_parameter_measure_with_unit
(*
```

Formal propositions:

WR1: The scattering_parameter_measure_with_unit shall not be any other type of measure_with_unit.

WR2: The unit_component shall be a ratio_unit.

5.2.3.1.328 seating_plane

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

The seating_plane is a shape_aspect that implements the ARM concept of Seating_plane (see 4.2.802).

EXPRESS specification:

```

*)
ENTITY seating_plane
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\seating_plane)) = 0;
END_ENTITY; -- seating_plane
( *

```

Formal propositions:

WR1: The seating_plane shall not be any other subtype of shape_aspect.

5.2.3.1.329 secondary_orientation_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A secondary_orientation_feature is a physical_unit_datum_feature that implements the ARM concept of Secondary_orientation_feature (see 4.2.803).

EXPRESS specification:

```

*)
ENTITY secondary_orientation_feature
  SUBTYPE OF (physical_unit_datum_feature);
WHERE

  WR1:SIZEOF ((TYPEOF (SELF)) *
    [ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRIMARY_ORIENTATION_FEATURE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SEATING_PLANE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'VIEWING_PLANE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT' ]) = 0;

  WR2: SELF.product_definitional = TRUE;

```

```

WR3: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_BOTTOM_SURFACE' IN
    TYPEOF (it.relate_shape_aspect))
    )) <= 1;

WR4: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_TOP_SURFACE' IN
    TYPEOF (it.relate_shape_aspect))
    )) <= 1;
END_ENTITY; -- secondary_orientation_feature
(*)

```

Formal propositions:

WR1: The secondary_orientation_feature shall not be a primary_orientation_feature, a seating_plane, a viewing_plane, a component_shape_aspect, or a physical_unit_datum.

WR2: A secondary_orientation_feature shall be on the surface of the product.

WR3: A primary_orientation_feature shall be associated with no more than one package_body_bottom_surface.

WR4: A primary_orientation_feature shall be associated with no more than one package_body_top_surface.

5.2.3.1.330 sequential_laminate_passage_based_fabrication_joint

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A sequential_laminate_passage_based_fabrication_joint is a shape_aspect that implements the ARM concept of Interconnect_module_product_assembly_view_fabrication_joint (see 4.2.458).

EXPRESS specification:

```

*)
ENTITY sequential_laminate_passage_based_fabrication_joint
    SUBTYPE OF (shape_aspect);
END_ENTITY; -- sequential_laminate_passage_based_fabrication_joint
(*)

```


5.2.3.1.331 shape_formed_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A shape_formed_terminal is an altered_package_terminal that implements the ARM concept of Shape_formed_terminal (see 4.2.817).

EXPRESS specification:

```
*)
ENTITY shape_formed_terminal
  SUBTYPE OF (altered_package_terminal);
WHERE

  WR1: (SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION' IN
    TYPEOF (ada.assigned_document)) AND
    (ada.assigned_document.kind\document_type.product_data_type =
    'lead form specification') AND
    (SIZEOF (QUERY (dr <* USEDIN (ada.assigned_document,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DOCUMENT_RELATIONSHIP.RELATED_DOCUMENT') |
    dr.relatng_document.kind\document_type.product_data_type =
    'material specification')) = 1))) = 1));
END_ENTITY; -- shape_formed_terminal
(*
```

Formal propositions:

WR1: The shape_formed_terminal shall have exactly one document_reference where the assigned_document is an ee_specification with a product_data_type of 'lead form specification' and is related to a document with a product_data_type of 'material specification'.

5.2.3.1.332 shape_modification

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A shape_modification is a shape_aspect that implements the ARM concepts of:

- Design_intent_modification (see 4.2.267);
- Electrical_isolation_removal_structure (see 4.2.343);

- Filled_area_material_removal_structure (see 4.2.385);
- Material_addition_feature (see 4.2.528);
- Material_removal_feature (see 4.2.533);
- Material_shape_modification (see 4.2.536);
- Thermal_isolation_removal_structure (see 4.2.903).

EXPRESS specification:

```

*)
ENTITY shape_modification
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF (QUERY (dim <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'design intent') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' IN
    TYPEOF (dim.relate_shape_aspect))) = 1;

  WR2: (SIZEOF (QUERY (pd<* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN
    TYPEOF (pd)) AND
    (pd\property_definition.description = 'modification causal')) = 1) OR
    (SIZEOF (QUERY (mcf <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'modification causal feature') |
    (SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_INTERFACE_TERMINAL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' ] *
    TYPEOF (mcf.relate_shape_aspect)) = 1) OR
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM' IN
    TYPEOF (mcf.relate_shape_aspect.of_shape.definition)))) = 1);

```

```

WR3: (NOT (SELF\shape_aspect.description IN ['electrical isolation removal',
'thermal isolation removal'])) OR (SIZEOF(
  QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (pd\property_definition.name = 'spacing requirement') AND
    (SIZEOF(QUERY (pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      (pdr\property_definition_relationship.name = 'spacing requirement') AND
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REQUIREMENTS_PROPERTY' IN
      TYPEOF(pdr.relatng_property_definition)))) = 1) ))
    = 1);

WR4: (NOT (SELF\shape_aspect.description IN ['electrical isolation removal',
'thermal isolation removal'])) OR
(SIZEOF (QUERY (di <* QUERY (sar <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  sar\shape_aspect_relationship.name = 'design intent') |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'STRATUM_FEATURE' IN
  TYPEOF (di.relatng_shape_aspect)) AND
  (di.relatng_shape_aspect\shape_aspect.description IN ['conductive filled area',
'connected filled area']))) = 1);

WR5: (NOT (SELF\shape_aspect.description = 'thermal isolation removal')) OR
(SIZEOF (QUERY (pd<* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REQUIREMENTS_PROPERTY' IN
  TYPEOF (pd)) AND
  (pd\property_definition.description = 'angular orientation requirement')))) = 1);

WR6: (NOT (SELF\shape_aspect.description = 'thermal isolation removal')) OR
(SIZEOF (QUERY (pd<* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REQUIREMENTS_PROPERTY' IN
  TYPEOF (pd)) AND
  (pd\property_definition.description = 'effective current capacity requirement')))) =

```

```

WR7: (NOT (SELF\shape_aspect.description = 'material addition feature')) OR
      (SIZEOF (QUERY (dim <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'material addition shape') |
        (dim.relatng_shape_aspect\shape_aspect.description =
        'material addition feature template')))) = 1);
END_ENTITY; -- shape_modification
( *

```

Formal propositions:

WR1: The shape_modification shall be related to exactly one stratum_feature as the 'design intent'.

WR2: The shape_modification shall have exactly one requirements_property with a description of 'modification causal', or it shall be related to exactly one component_shape_aspect, component_terminal, component_interface_terminal, stratum_feature, or shape_aspect of a stratum as the 'modification causal feature'.

WR3: If the shape_modification has a description of 'electrical isolation removal' or 'thermal isolation removal', it shall have exactly one requirements_property with a description of 'spacing requirement'.

WR4: If the shape_modification has a description of 'electrical isolation removal' or 'thermal isolation removal', it shall be related to exactly one stratum_feature with a description of 'conductive filled area' or 'connected filled area' as the 'design intent'.

WR5: If the shape_modification has a description of 'thermal isolation removal', it shall have exactly one requirements_property with the description of 'angular orientation requirement'.

WR6: If the shape_modification has a description of 'thermal isolation removal', it shall have exactly one requirements_property with the description of 'effective current capacity requirement'.

WR7: If the shape_modification has a description of 'material addition feature', it shall be related to exactly one 'material addition feature template'.

5.2.3.1.333 shape_representation_relationship_with_transformation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A shape_representation_relationship_with_transformation is a shape_representation_relationship and a representation_relationship_with_transformation that implements the ARM concept of Component_part_2d_non_planar_geometric_representation_relationship (see 4.2.132).

EXPRESS specification:

```

*)
ENTITY shape_representation_relationship_with_transformation
  SUBTYPE OF (shape_representation_relationship, representation_relationship_with_transformation)
END_ENTITY; -- shape_representation_relationship_with_transformation
( *

```

5.2.3.1.334 shell_based_2d_wireframe_shape_representation

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A shell_based_2d_wireframe_shape_representation is a shape_representation that implements the ARM concept of Shell_based_2d_wireframe (see 4.2.819).

EXPRESS specification:

```

*)
ENTITY shell_based_2d_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
WHERE

  WR1: SIZEOF (QUERY (it <* SELF\representation.items |
    NOT (SIZEOF (
      [ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D' ] *
      TYPEOF (it)) = 1))) = 0;

  WR2: SIZEOF (QUERY (it <* SELF\representation.items |
    SIZEOF (
      [ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM' ] *
      TYPEOF (it)) = 1)) >= 1;

```

```

WR3: sizeof (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
typeof (it)) |
NOT (sizeof (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN typeof (sb)) |
NOT (sizeof (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN typeof (wsb)) |
NOT (sizeof (QUERY (el <* eloop\path.edge_list |
NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_CURVE' IN
typeof (el.edge_element)))) = 0))) = 0))) = 0))) = 0;

WR4: sizeof (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
typeof (it)) |
NOT (sizeof (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN typeof (sb)) |
NOT (sizeof (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN typeof (wsb)) |
NOT (sizeof (QUERY (pline_el <* QUERY (el <* eloop\path.edge_list |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE' IN
typeof (el.edge_element\edge_curve.edge_geometry)) |
NOT (sizeof (pline_el.edge_element\edge_curve.
edge_geometry\polyline.points) > 2))) = 0))) = 0))) = 0;

```

```

WR5: sizeof (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
typeof (it)) |
NOT (sizeof (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN typeof (sb)) |
NOT (sizeof (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN typeof (wsb)) |
NOT (sizeof (QUERY (el <* eloop\path.edge_list |
NOT (valid_2d_wireframe_edge_curve
(el.edge_element\edge_curve.edge_geometry,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')))) =
0))) = 0))) = 0))) = 0;

WR6: sizeof (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
typeof (it)) |
NOT (sizeof (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN typeof (sb)) |
NOT (sizeof (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN typeof (wsb)) |
NOT (sizeof (QUERY (el <* eloop\path.edge_list |
NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
typeof (el.edge_element.edge_start)) AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
typeof (el.edge_element.edge_end))) )) = 0))) = 0))) = 0))) = 0;

```

```

WR7: SIZEOF (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
  TYPEOF (it)) |
NOT (SIZEOF (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN TYPEOF (sb)) |
NOT (SIZEOF (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN TYPEOF (wsb)) |
NOT (SIZEOF (QUERY (el <* eloop\path.edge_list |
NOT ((valid_wireframe_vertex_point
(el.edge_element.edge_start\vertex_point.vertex_geometry)) AND
(valid_wireframe_vertex_point
(el.edge_element.edge_end\vertex_point.vertex_geometry)))))) =
0))) = 0))) = 0))) = 0;

WR8: SIZEOF (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
  TYPEOF (it)) |
NOT (SIZEOF (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN TYPEOF (sb)) |
NOT (SIZEOF (QUERY (eloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP' IN TYPEOF (wsb)) |
NOT (SIZEOF (QUERY (con_edges <* QUERY (el <* eloop\path.edge_list |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC' IN
  TYPEOF (el.edge_element\edge_curve.edge_geometry)) |
NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_2D' IN
  TYPEOF (con_edges.edge_element\edge_curve.
edge_geometry\conic.position)))))) = 0))) = 0))) = 0))) = 0;

```



```

WR9: SIZEOF (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
TYPEOF(it)) |
NOT (SIZEOF (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN TYPEOF (sb)) |
NOT (SIZEOF (QUERY (vloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP' IN TYPEOF (wsb)) |
NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
TYPEOF (vloop\vertex_loop.loop_vertex)))) = 0))) = 0))) = 0;

WR10: SIZEOF (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
TYPEOF(it)) |
NOT (SIZEOF (QUERY (ws <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL' IN TYPEOF (sb)) |
NOT (SIZEOF (QUERY (vloop <* QUERY (wsb <*
ws\wire_shell.wire_shell_extent |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP' IN TYPEOF (wsb)) |
NOT (valid_wireframe_vertex_point (vloop\vertex_loop.
loop_vertex\vertex_point.vertex_geometry)))) =
0))) = 0))) = 0;

WR11: SIZEOF (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL' IN
TYPEOF(it)) |
NOT (SIZEOF (QUERY (vs <* QUERY (sb <*
sbwm\shell_based_wireframe_model.sbwm_boundary |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_SHELL' IN TYPEOF (sb)) |
NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT' IN
TYPEOF (vs\vertex_shell.vertex_shell_extent.loop_vertex))))
= 0))) = 0;

```

```

WR12: sizeof (QUERY (sbwm <* QUERY (it <* SELF\representation.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL' IN
    TYPEOF(it)) |
    NOT (sizeof (QUERY (vs <* QUERY (sb <*
        sbwm\shell_based_wireframe_model.sbwm_boundary |
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VERTEX_SHELL' IN TYPEOF (sb)) |
        NOT (valid_wireframe_vertex_point (vs\vertex_shell.
            vertex_shell_extent.loop_vertex\vertex_point.vertex_geometry))))
    = 0))) = 0;

WR13: sizeof (QUERY (mi <* QUERY (it <* SELF\representation.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM' IN TYPEOF (it)) |
    NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION' IN
    TYPEOF(mi\mapped_item.mapping_source.mapped_representation)
    ))) = 0;

WR14: SELF.context_of_items\geometric_representation_context.
    coordinate_space_dimension = 2;
END_ENTITY; -- shell_based_2d_wireframe_shape_representation
(*

```

Formal propositions:

WR1: The items in a shell_based_2d_wireframe_shape_representation shall be a shell_based_wireframe_model, a mapped_item, or an axis2_placement_2d.

WR2: At least one of the items in a shell_based_2d_wireframe_shape_representation shall be either a shell_based_wireframe_model or a mapped_item.

WR3: Every edge defined for an edge_loop in a shell_based_wireframe_model shall be an edge_curve.

WR4: Every polyline that underlies an edge in a shell_based_wireframe_model shall contain more than two distinct points.

WR5: The edge_geometry that underlies an edge for a shell_based_wireframe_model shall be a line, circle, ellipse, parabola, hyperbola, b_spline_curve, offset_curve_2d, polyline, or curve_replica and the curves that have a basis defined by other curves shall be defined consistently.

WR6: Every vertex defined as the start or end vertex for an edge in a shell_based_wireframe_model shall be a vertex_point.

WR7: The vertex_geometry that underlies the vertices that define the boundaries of the edges in an edge_loop for a shell_based_wireframe_model shall be a cartesian_point or point_replica, and the point_replica shall replicate either another point_replica or a cartesian_point.

WR8: The position of a conic that underlies an edge in an edge_loop for a shell_based_wireframe_model shall only be an axis2_placement_2d.

WR9: The vertex that defines the vertex_loop which is used as a bound in a shell_based_wireframe_model shall be a vertex_point.

WR10: The vertex that defines the vertex_loop which is used as a bound in a shell_based_wireframe_model shall be underlaid by a cartesian_point or point_replica, and the point_replica shall replicate either another point_replica or a cartesian_point.

WR11: The vertex that defines the vertex_loop which is used as the vertex_shell_extent for a vertex_shell in a shell_based_wireframe_model shall be a vertex_point.

WR12: The vertex that defines the vertex_loop which is used as the vertex_shell_extent for a vertex_shell in a shell_based_wireframe_model shall be underlaid by a cartesian_point or point_replica, and the point_replica shall replicate either another point_replica or a cartesian_point.

WR13: If there is a mapped_item in a shell_based_2d_wireframe_shape_representation, the source of the mapped_item shall be a shell_based_2d_wireframe_shape_representation.

WR14: The dimensionality of the shell_based_2d_wireframe_shape_representation is two.

5.2.3.1.335 signal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A signal is a characterized_object that implements the ARM concept of Signal (see 4.2.823).

EXPRESS specification:

```
*)
ENTITY signal
  SUBTYPE OF (characterized_object);
WHERE

  WR1: SIZEOF (QUERY (aca <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SIGNAL_CATEGORY' IN
    TYPEOF (aca.assigned_group))) >= 1;

  WR2: (SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    SIZEOF (QUERY (duc <* USEDIN (ada.assigned_document,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    duc\document_usage_constraint.subject_element = 'signal category')) = 1)) = 1);
```

```

    WR3: SIZEOF (USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1;
END_ENTITY; -- signal
(*)

```

Formal propositions:

WR1: The signal shall be classified in at least one signal_category.

WR2: The signal shall be referenced by exactly one document that has a usage constraint that identifies the 'signal category'.

WR3: The signal shall be referenced by exactly one document.

5.2.3.1.336 signal_category

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A signal_category is a group and an externally_defined_item that specifies the External_definition (see 4.2.361) for a Signal_property (see 4.2.824).

EXPRESS specification:

```

*)
ENTITY signal_category
    SUBTYPE OF (group, externally_defined_item);
WHERE

    WR1: SELF\group.description IN ['signal characteristic category',
        'signal property category'];
END_ENTITY; -- signal_category
(*)

```

Formal propositions:

WR1: The signal_category shall have a group description of either 'signal characteristic category' or 'signal property category'.

5.2.3.1.337 solid_curve_font

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A solid_curve_font is a pre_defined_curve_font that implements the ARM concept of Solid_curve (see 4.2.831).

EXPRESS specification:

```

*)
ENTITY solid_curve_font
  SUBTYPE OF (pre_defined_curve_font);
END_ENTITY; -- solid_curve_font
( *

```

5.2.3.1.338 start_request

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A start_request is a formal notification of a desire to begin a new design for a product or group of **products**.

EXPRESS specification:

```

*)
ENTITY start_request
  SUBTYPE OF (action_request_assignment);
    items : SET [1:?] OF start_request_item;
END_ENTITY; -- start_request
( *

```

5.2.3.1.339 start_work

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A start_work identifies the **start_requests** that have been fulfilled and, as a result, have established a new version of a product.

EXPRESS specification:

```

*)
ENTITY start_work
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF work_item;
WHERE

    WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DIRECTED_ACTION' IN
        TYPEOF (SELF.assigned_action);
END_ENTITY; -- start_work
( *

```

Formal propositions:

WR1: The start_work shall assign a directed_action.

5.2.3.1.340 statistical_dimensional_location

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A statistical_dimensional_location is a dimensional_location that implements the combination of the ARM concepts Statistical_dimensional_tolerance (see 4.2.840) and Dimensional_location (see 4.2.300).

EXPRESS specification:

```
*)
ENTITY statistical_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY; -- statistical_dimensional_location
(*
```

Informal propositions:

IP1: The statistical_dimensional_location shall have exactly one property_definition with a description of 'dimensional location property', and that property_definition shall be related to at most one property_definition, of a dimensional_location, with a description of 'dimensional location property' as the 'substitutable dimension'.

IP2: The statistical_dimensional_location shall have exactly one property_definition with a description of 'dimensional location property', and that property_definition shall relate to a requirements_property to specify the 'dimensional tolerance statistical control requirement'.

IP3: The statistical_dimensional_location shall have exactly one property_definition with a description of 'dimensional location property', and if that property_definition is related to a property_definition, of a dimensional_location, with a description of 'dimensional location property' as the 'substitutable dimension', that dimensional_location shall not be a statistical_dimensional_location.

IP4: The statistical_dimensional_location shall have exactly one property_definition with a description of 'dimensional location property', and if that property_definition is related to a property_definition, of a dimensional_location, with a description of 'dimensional location property' as the 'substitutable dimension', both the statistical_dimensional_location and the dimensional_location shall specify the same two shape_aspects as the related_shape_aspect and the relating_shape_aspect.

5.2.3.1.341 statistical_dimensional_size

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `statistical_dimensional_size` is a `dimensional_size_property` that implements the combination of the ARM concepts `Statistical_dimensional_tolerance` (see 4.2.840) and `Dimensional_size` (see 4.2.301).

EXPRESS specification:

```

*)
ENTITY statistical_dimensional_size
  SUBTYPE OF (dimensional_size_property);
WHERE

  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_SIZE' IN
    TYPEOF (pdr.related_property_definition)) AND
    (pdr\property_definition_relationship.name = 'substitutable dimension'))
    <= 1))) <= 1;

  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN
    TYPEOF (pdr.related_property_definition)) AND
    (pdr\property_definition_relationship.name =
    'dimensional tolerance statistical control requirement'))
    <= 1))) <= 1;

```

```

WR3: (NOT(SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
(SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' DIMENSIONAL_SIZE' IN
TYPEOF (pdr.related_property_definition)) AND
(pdr\property_definition_relationship.name = 'substitutable dimension'))
<= 1))) <= 1))
OR (NOT(SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
(SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' DIMENSIONAL_SIZE' IN
TYPEOF (pdr.related_property_definition)) AND
(NOT( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' STATISTICAL_DIMENSIONAL_SIZE' IN
TYPEOF (pdr.related_property_definition))) AND
(pdr\property_definition_relationship.name = 'substitutable dimension'))
<= 1))) <= 1));
END_ENTITY; -- statistical_dimensional_size
(*

```

Formal propositions:

WR1: The statistical_dimensional_size shall be related to at most one dimensional_size as the 'substitutable dimension'.

WR2: The statistical_dimensional_size shall relate to a requirements_property to specify the 'dimensional tolerance statistical control requirement'.

WR3: If the statistical_dimensional_size is related to a dimensional_size, as the 'substitutable dimension', that dimensional_size shall not be a statistical_dimensional_size.

Informal propositions:

IP1: If the statistical_dimensional_size is related to a dimensional_size as the 'substitutable dimension', both the statistical_dimensional_size and the dimensional_size shall apply to the same shape_aspect.

5.2.3.1.342 statistical_geometric_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A statistical_geometric_tolerance is a physical_unit_geometric_tolerance that implements the ARM concept of Statistical_geometric_tolerance (see 4.2.841).

EXPRESS specification:

```

*)
ENTITY statistical_geometric_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
WHERE

  WR1: SIZEOF(QUERY(gtr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE' IN
    TYPEOF(gtr.relatng_geometric_tolerance)) AND
    (gtr\geometric_tolerance_relationship.name = 'substitutable geometric tolerance')
  )) <= 1;

  WR2: SIZEOF(QUERY(pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN
    TYPEOF(pdr.related_property_definition)) AND
    (pdr\property_definition_relationship.name = 'geometric tolerance statistical control requirement')
  )) = 1;

  WR3: SIZEOF(QUERY(gtr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STATISTICAL_GEOMETRIC_TOLERANCE' IN
    TYPEOF(gtr.relatng_geometric_tolerance)) AND
    (gtr\geometric_tolerance_relationship.name = 'substitutable geometric tolerance')
  )) <= 0;
END_ENTITY; -- statistical_geometric_tolerance
(

```

Formal propositions:

WR1: The statistical_geometric_tolerance shall be related to at most one physical_unit_geometric_tolerance as the 'substitutable geometric tolerance'.

WR2: The statistical_geometric_tolerance shall relate to exactly one requirements_property to specify the 'geometric tolerance statistical control requirement'.

WR3: The statistical_geometric_tolerance shall be related to exactly zero statistical_geometric_tolerance as the 'substitutable geometric tolerance'.

Informal propositions:

IP1: If the statistical_geometric_tolerance is related to exactly one physical_unit_geometric_tolerance as the 'substitutable geometric tolerance', the statistical_geometric_tolerance and the physical_unit_geometric_tolerance shall specify the same shape_aspect as their toleranced_shape_aspect.

5.2.3.1.343 stratum

A stratum is a product_definition that implements the ARM concepts of:

- Design_layer_stratum (see 4.2.275);
- Documentation_layer_stratum (see 4.2.306);
- Stratum (see 4.2.845).

EXPRESS specification:

```

*)
ENTITY stratum
  SUBTYPE OF (product_definition);
  UNIQUE
    UR1: SELF\product_definition.id, SELF\product_definition.formation;
  WHERE

    WR1: EXISTS(SELF\product_definition.name);

    WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
    pdr\property_definition_relationship.name = 'technology usage') |
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM_TECHNOLOGY' IN
    TYPEOF (tu.relatng_property_definition.definition))) = 1)) = 1;

```

```

WR3: (SIZEOF (QUERY (ada <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' APPLIED_DOCUMENT_REFERENCE.ITEMS') |
SIZEOF (QUERY (duc <* USEDIN (ada.assigned_document,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
duc\document_usage_constraint.subject_element = 'attachment region size')) = 1)) <=
(SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'stratum usage') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' EXTERNAL_DEFINITION' IN
TYPEOF (tu.relatng_property_definition))) = 1)) <= 1));

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation)) |
sr_pdr.used_representation\representation.name = '3d bound volume shape')) <= 1)))
= 0;

WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (sr_pdr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_REPRESENTATION' IN
TYPEOF (pdr.used_representation)) |
sr_pdr.used_representation\representation.name =
'planar projected shape')) <= 1))) = 0;

WR6: SIZEOF (QUERY (acu <* QUERY (pdr <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' ASSEMBLY_COMPONENT_USAGE' IN
TYPEOF (pdr)) |
acu\product_definition_relationship.name = 'interconnect module stratum assembly re
)) >= 1;

```

```

WR7: NOT(SELf\product_definition.name = 'design layer') OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'technology usage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY' IN
TYPEOF (tu.relatinq_property_definition.definition)) AND
(tu.relatinq_property_definition.definition\characterized_object.description =
'design layer')))) = 1)) = 1);

WR8: NOT(SELf\product_definition.name = 'design layer') OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (sa <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') |
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAYER_CONNECTION_POINT' IN
TYPEOF (sa))) >= 1)) >= 1);

WR9: NOT(SELf\product_definition.name = 'documentation layer') OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
SIZEOF (QUERY (tu <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
pdr\property_definition_relationship.name = 'technology usage') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY' IN
TYPEOF (tu.relatinq_property_definition.definition)) AND
(tu.relatinq_property_definition.definition\characterized_object.description =
'documentation layer')))) = 1)) = 1);

```

```

WR10: NOT(SELF\product_definition.name = 'design layer') OR
      ((NOT(SELF\product_definition.description = 'primary design layer stratum') OR
        (SIZEOF(QUERY(imps <* USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF (QUERY (sa <* USEDIN (imps,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        SIZEOF (QUERY (sar <* USEDIN (sa,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT' IN
        TYPEOF (sar.relatng_shape_aspect))) AND
        (sar.relatng_shape_aspect\shape_aspect.description =
          'interconnect module primary surface')))) >= 0)) >= 0)) = 1)) OR
      (NOT (SELF\product_definition.description = 'non primary design layer stratum') OR
        (SIZEOF(QUERY(imps <* USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF (QUERY (sa <* USEDIN (imps,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        SIZEOF(QUERY (sar <* USEDIN (sa,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'SHAPE_ASPECT' IN
        TYPEOF (sar.relatng_shape_aspect))) AND
        (sar.relatng_shape_aspect\shape_aspect.description =
          'interconnect module primary surface')))) >= 0)) >= 0)) = 0)))));
END_ENTITY; -- stratum
(*

```

Formal propositions:

UR1: The name shall be unique in the context of the version of the product.

WR1: The name shall exist.

WR2: The stratum shall have exactly one property_definition that is related to a property_definition of a stratum_technology as the 'technology usage'.

WR3: The stratum shall be referenced by at most one document with a usage constraint that specifies the 'attachment region size', or it shall have at most one property that is related to an external_definition as the 'stratum usage'.

WR4: The stratum shall have at most one shape_representation with a name of '3d bound volume shape'.

WR5: The stratum shall have at most one shape_representation with a name of 'planar projected shape'.

WR6: The stratum shall be the component in at least one assembly_component_usage as the 'interconnect module stratum assembly relationship'.

WR7: If the stratum is a definition of a product that has a name of 'design layer', it shall have exactly one property_definition that is related to exactly one property_definition of a stratum_technology with a description of 'design layer' as the 'technology usage'.

WR8: If the stratum is a definition of a product that has a name of 'design layer', it shall have at least one layer_connection_point.

WR9: If the stratum is a definition of a product that has a name of 'documentation layer', it shall have exactly one property_definition that is related to exactly one property_definition of a stratum_technology with a description of 'documentation layer' as the 'technology usage'.

WR10: If the stratum is a definition of a product that has a name of 'design layer', then either the description shall be 'primary design layer stratum' and it shall be assigned to an interconnect module primary surface, or the description shall be 'non primary design layer stratum' and it shall not be assigned to an interconnect module primary surface.

5.2.3.1.344 stratum_concept_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_concept_relationship is a shape_aspect and a shape_aspect_relationship that implements the ARM concepts of:

- Dielectric_crossover_area (see 4.2.288);
- Inter_stratum_join_implementation (see 4.2.436);
- Physical_network_supporting_stratum_feature_conductive_join (see 4.2.662);
- Stratum_concept_relationship (see 4.2.849);
- Stratum_feature_conductive_join (see 4.2.853).

EXPRESS specification:

```
*)
ENTITY stratum_concept_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
WHERE
```

```

WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY' IN
    TYPEOF (pd))) >= 1;

WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    SIZEOF (USEDIN (pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) = 1;

WR3: (NOT (SELF\shape_aspect_relationship.name =
    'dielectric crossover area')) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' IN
    TYPEOF (SELF.relatng_shape_aspect)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE' IN
    TYPEOF (SELF.related_shape_aspect)));

WR4: (NOT (SELF\shape_aspect_relationship.name =
    'dielectric crossover area')) OR
    (SIZEOF (QUERY (rdc <* QUERY (sar <* USEDIN (SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        sar\shape_aspect_relationship.name = 'resulting dielectric crossover') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE' IN
        TYPEOF (rdc.relatng_shape_aspect)))) = 1);

WR5: (NOT (SELF\shape_aspect_relationship.name =
    'stratum feature conductive join')) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND' IN
    TYPEOF (SELF.relatng_shape_aspect)) OR
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT' IN
    TYPEOF (SELF.relatng_shape_aspect)) AND
    (SELF.relatng_shape_aspect\shape_aspect.description =
    'stratum feature template component')));

```

```

WR6: (NOT (SELF\shape_aspect_relationship.name =
'stratum feature conductive join')) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND' IN
TYPEOF (SELF.related_shape_aspect)) OR
(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT' IN
TYPEOF (SELF.related_shape_aspect)) AND
(SELF.related_shape_aspect\shape_aspect.description =
'stratum feature template component')));

WR7: (NOT (SELF\shape_aspect_relationship.name =
'stratum feature conductive join')) OR
(SIZEOF (QUERY (fj <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'features join') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE' IN
TYPEOF (fj.related_shape_aspect)) AND
(fj.related_shape_aspect\shape_aspect.description =
'bonded conductive base blind via')))) <= 1);

WR8: (NOT ((SELF\shape_aspect.description =
'physical network supporting stratum feature conductive join') AND
(SELF\shape_aspect_relationship.name =
'stratum feature conductive join')) OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT' IN
TYPEOF (ji.relateing_shape_aspect)) AND
(ji.relateing_shape_aspect\shape_aspect.name =
'inter stratum join')))) = 1);
END_ENTITY; -- stratum_concept_relationship
( *

```

Formal propositions:

WR1: The stratum_concept_relationship shall have at least one requirements_property.

WR2: The stratum_concept_relationship shall have exactly one property_definition that is represented.

WR3: If the stratum_concept_relationship has a shape_aspect_relationship name of 'dielectric crossover area', it shall relate two stratum_features.

WR4: If the `stratum_concept_relationship` has a `shape_aspect_relationship` name of 'dielectric crossover area', the `stratum_concept_relationship` shall be related to exactly one `stratum_feature` as a 'resulting dielectric crossover'.

WR5: If the `stratum_concept_relationship` has a `shape_aspect_relationship` name of 'stratum feature conductive join', it shall have a `relating_shape_aspect` that is either a `land` or a `component_definition` with a description of 'stratum feature template component'.

WR6: If the `stratum_concept_relationship` has a `shape_aspect_relationship` name of 'stratum feature conductive join', it shall have a `related_shape_aspect` that is either a `land` or a `component_definition` with a description of 'stratum feature template component'.

WR7: If the `stratum_concept_relationship` has a `shape_aspect_relationship` name of 'stratum feature conductive join', the `stratum_concept_relationship` shall relate to at most one `plated_passage` that has a description of 'bonded conductive base blind via' to specify the 'features join'.

WR8: If the `stratum_concept_relationship` has a `shape_aspect` description of 'physical network supporting stratum feature conductive join' and a `shape_aspect_relationship` name of 'stratum feature conductive join', the `stratum_concept_relationship` shall be related, as the 'join implementation', to exactly one `join_shape_aspect` that has a name of 'inter stratum join'.

5.2.3.1.345 `stratum_feature`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `stratum_feature` is a `shape_aspect` that implements the ARM concepts of:

- `Conductive_filled_area` (see 4.2.170);
- `Conductor` (see 4.2.176);
- `Connected_filled_area` (see 4.2.184);
- `Stratum_feature` (see 4.2.851).

EXPRESS specification:

```
*)
ENTITY stratum_feature
  SUPERTYPE OF (fiducial_stratum_feature)
  SUBTYPE OF (shape_aspect);
WHERE
```

```
WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM' IN
      TYPEOF (SELF.of_shape.definition);
```

```

WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
SIZEOF (QUERY (it <* pdr.used_representation.items |
( ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' DESCRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF (it)) AND
(it\representation_item.name = 'feature of size') AND
(it\descriptive_representation_item.description IN
[ 'true', 'false']))) = 1)) = 1))) = 0;

WR3: (NOT(EXISTS(SELF\shape_aspect.description))) OR
(NOT (SELF\shape_aspect.description = 'conductor') OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
( ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' JOIN_SHAPE_ASPECT' IN
TYPEOF (ji.relating_shape_aspect)) AND
(ji.relating_shape_aspect\shape_aspect.name =
'intra stratum join')) = 1))) = 1));

WR4: (NOT(EXISTS(SELF\shape_aspect.description))) OR
(NOT (SELF\shape_aspect.description = 'connected filled area') OR
(SIZEOF (QUERY (ji <* QUERY (sar <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
sar\shape_aspect_relationship.name = 'join implementation') |
( ' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' JOIN_SHAPE_ASPECT' IN
TYPEOF (ji.relating_shape_aspect)) AND
(ji.relating_shape_aspect\shape_aspect.name =
'intra stratum join')) = 1))) = 1));

WR5: (NOT(EXISTS(SELF\shape_aspect.description))) OR
(NOT (SELF\shape_aspect.description = 'conductor') OR
(SELF\shape_aspect.of_shape.definition.name = 'design layer'));

WR6: (NOT(EXISTS(SELF\shape_aspect.description))) OR
(NOT (SELF\shape_aspect.description = 'connected filled area') OR
(SELF\shape_aspect.of_shape.definition.name = 'design layer'));
END_ENTITY; -- stratum_feature
( *

```

Formal propositions:

WR1: The stratum_feature shall be an aspect of the shape of a stratum.

WR2: The stratum_feature shall have exactly one representation that specifies if the stratum_feature is a 'feature of size'.

WR3: If the stratum_feature has a description of 'conductor', it shall be related to exactly one join_shape_aspect with a name of 'intra stratum join' as the 'join implementation'.

WR4: If the stratum_feature has a description of 'connected filled area', it shall be related to exactly one join_shape_aspect with a name of 'intra stratum join' as the 'join implementation'.

WR5: If the stratum_feature has a description of 'conductor', it shall be related to a 'design layer'.

WR6: If the stratum_feature has a description of 'connected filled area ', it shall be related to a 'design layer'.

5.2.3.1.346 stratum_occurrence_specific_padstack_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_occurrence_specific_padstack_definition is a product_definition that implements the ARM concepts of: Stratum_occurrence_specific_padstack_definition (see 4.2.866)

EXPRESS specification:

```
*)
ENTITY stratum_occurrence_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- stratum_occurrence_specific_padstack_definition
(*
```

5.2.3.1.347 stratum_surface

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_surface is a shape_aspect that implements the ARM concepts of Stratum_surface (see 4.2.868) and Stratum_average_surface (see 4.2.847).

EXPRESS specification:

```
*)
ENTITY stratum_surface
  SUBTYPE OF (shape_aspect);
WHERE
```

```

WR1: SELF\shape_aspect.description IN ['primary surface',
    'secondary surface', 'average surface'];

WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM' IN TYPEOF(SELF\shape_aspect.of_shape.definition);
END_ENTITY; -- stratum_surface
(*

```

Formal propositions:

WR1: The stratum_surface shall have a description of 'primary surface', 'secondary surface', or 'average surface'.

WR2: The stratum_surface shall be the shape of a stratum.

5.2.3.1.348 stratum_technology

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_technology is a characterized_object that implements the ARM concepts of:

- Design_layer_technology (see 4.2.276);
- Documentation_layer_technology (see 4.2.307);
- Stratum_technology (see 4.2.871).

EXPRESS specification:

```

*)
ENTITY stratum_technology
    SUBTYPE OF (characterized_object);
UNIQUE
    UR1: SELF\characterized_object.name;
WHERE

WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_DESIGNATION.DEFINITIONS')) = 1;

```

```

WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'stiffness class representation')) <= 1))) = 0;

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation')) = 1))) = 1;

WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (scr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name = 'stratum class representation') |
    NOT (SIZEOF (QUERY (it <* scr.used_representation.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (it)) AND
    (it\representation_item.name = 'laminate stiffness class') AND
    (it\descriptive_representation_item.description IN ['fluid like', 'conformal coat',
    'stiff laminate'])))) = 1))) = 0))) = 0;

WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT ({2 <= SIZEOF (pcr.used_representation.items)
    <= 8}))) = 0))) = 0;

```

```

WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
' physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
NOT (it\representation_item.name IN ['layer purpose',
'layer position',
'maximum feature size requirement',
'minimum finished feature spacing',
'minimum finished feature size',
'maximum thickness',
'minimum thickness'])))) = 0))) = 0))) = 0;

```

```

WR7: NOT(EXISTS(SELF\characterized_object.description)) OR
(NOT (SELF\characterized_object.description = 'design layer') OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
' physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
[' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MEASURE_REPRESENTATION_ITEM',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name = 'minimum finished feature size')) = 1))) = 0))) = 0;

```

```

WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      (SIZEOF (
        ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it\representation_item.name = 'maximum thickness')))) = 1))) = 0))) = 0;

WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      (SIZEOF (
        ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it\representation_item.name = 'minimum thickness')))) = 1))) = 0))) = 0;

WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    pdr.used_representation\representation.name =
    'physical characteristics representation') |
    NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      (SIZEOF (
        ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it\representation_item.name = 'maximum feature size requirement')))) <= 1))) = 0)))

```

```

WR11: SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name = 'minimum finished feature spacing')) <= 1))) = 0)))

WR12: NOT(EXISTS(SELF\characterized_object.description)) OR
((NOT (SELF\characterized_object.description = 'documentation layer')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
(it\representation_item.name = 'layer purpose')) = 1))) = 0))) = 0));

WR13: NOT(EXISTS(SELF\characterized_object.description)) OR
((NOT (SELF\characterized_object.description = 'design layer')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
pdr.used_representation\representation.name =
'physical characteristics representation') |
NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
(SIZEOF (
['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it\representation_item.name = 'minimum finished feature spacing')) = 1))) = 0))) =

```



```

WR14: NOT(EXISTS(SELF\characterized_object.description)) OR
      ((NOT (SELF\characterized_object.description = 'design layer')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation\representation.name =
      'physical characteristics representation') |
      NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND
      (it\representation_item.name = 'layer purpose') AND
      (it\descriptive_representation_item.description IN ['power or ground', 'other signal
      'lands only'])))) = 1))) = 0))) = 0));

WR15: SIZEOF (QUERY (pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (pcr <* QUERY (pdr <* USEDIN (pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      pdr.used_representation\representation.name =
      'physical characteristics representation') |
      NOT (SIZEOF (QUERY (it <* pcr.used_representation.items |
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND
      (it\representation_item.name = 'layer position') AND
      (it\descriptive_representation_item.description IN [
      'primary',
      'secondary',
      'internal',
      'external',
      'all'])))) = 1))) = 0))) = 0;

WR16: NOT(EXISTS(SELF\characterized_object.description)) OR
      (SELF\characterized_object.description IN [
      'design layer',
      'documentation layer']);
END_ENTITY; -- stratum_technology
( *

```

Formal propositions:

UR1: The name shall be unique.

WR1: The stratum_technology shall have exactly one material designation.

WR2: The stratum_technology shall have at most one representation with a name of 'stiffness class representation'.

WR3: The stratum_technology shall have exactly one representation with a name of 'physical characteristics representation'.

WR4: Exactly one item in the representation with a name of 'stiffness class representation' shall be a descriptive_representation_item with a name of 'laminate stiffness class' and a description of 'fluid like', 'conformal coat', or 'stiff laminate'.

WR5: The stratum_technology shall have at least two and at most eight items in its representation with a name of 'physical characteristics representation'.

WR6: The items in the representation with a name of 'physical characteristics representation' of a stratum_technology shall have a name of either 'documentation layer purpose', 'design layer purpose', 'design layer position', 'maximum feature size requirement', 'minimum finished feature spacing', 'minimum finished feature size', 'maximum stratum thickness', or 'minimum stratum thickness'.

WR7: Exactly one item in the representation with a name of 'physical characteristics representation' of a stratum_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum finished feature size'.

WR8: Exactly one item in the representation with a name of 'physical characteristics representation' of a stratum_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'maximum stratum thickness'.

WR9: Exactly one item in the representation with a name of 'physical characteristics representation' of a stratum_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum stratum thickness'.

WR10: At most one item in the representation with a name of 'physical characteristics representation' of a stratum_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'maximum feature size requirement'.

WR11: At most one item in the representation with a name of 'physical characteristics representation' of a stratum_technology shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum finished feature spacing'.

WR12: If the stratum_technology has a description of 'documentation layer', exactly one item in its representation with a name of 'physical characteristics representation' shall be a descriptive_representation_item with a name of 'layer purpose'.

WR13: If the stratum_technology has a description of 'design layer', exactly one item in its representation with a name of 'physical characteristics representation' shall be a measure_representation_item and a length_measure_with_unit with a name of 'minimum finished feature spacing'.

WR14: If the stratum_technology has a description of 'design layer', exactly one item in its representation with a name of 'physical characteristics representation' shall be a descriptive_representation_item with a name of 'layer purpose' and a description of 'power or ground', 'other signal', or 'lands only'.

WR15: If the stratum_technology has a description of 'design layer', exactly one item in its representation with a name of 'physical characteristics representation' shall be a descriptive_representation_item with a name of 'layer position' and a description of 'primary', 'secondary', 'all', 'external' or 'internal'.

WR16: If the description attribute exists, it shall be either 'design layer' or 'documentation layer'.

5.2.3.1.349 stratum_technology_link

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_technology_link is a property_definition_relationship that implements the ARM concept of Stratum_technology_link (see 4.2.872)

EXPRESS specification:

```

*)
ENTITY stratum_technology_link
  SUPERTYPE OF ( printed_part_template_link )
  SUBTYPE OF ( property_definition, property_definition_relationship );
WHERE

  WR1: SELF\property_definition_relationship.related_property_definition.definition :<>:
        SELF\property_definition_relationship.relateing_property_definition.definition;

  WR2: ( SIZEOF( TYPEOF( SELF ) - TYPEOF( SELF\property_definition_relationship ||
        SELF\stratum_technology_link || SELF\property_definition ) ) = 0 ) OR
        ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRINTED_PART_TEMPLATE_LINK' IN TYPEOF( SELF ) );

  WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_TECHNOLOGY' IN
        TYPEOF( SELF\property_definition_relationship.related_property_definition.definition );

  WR4: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_TECHNOLOGY' IN
        TYPEOF( SELF\property_definition_relationship.relateing_property_definition.definition );

  WR5: SELF\property_definition.name = '';

  WR6: SELF\property_definition.description = '';

  WR7: SELF\property_definition_relationship.name = '';

  WR8: SELF\property_definition_relationship.description = '';

```

```

WR9: ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION' IN TYPEOF(SELF\property_definition.definition)) AND
      (SELF\property_definition.definition.description = 'printed part template');

WR10: SIZEOF (QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
      pdr.name = 'precedent stratum technology link')) <= 1;

WR11: SIZEOF (QUERY (pdr <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |
      pdr.name = 'precedent stratum technology link')) <= 1;
END_ENTITY; -- stratum_technology_link
(*

```

Formal propositions:

WR1: The stratum_technology_link shall not relate to the same property_definition definition through the related and relating attributes.

WR2: The stratum_technology_link shall not be any other subtype of property_definition_relationship or property_definition.

WR3: The stratum_technology_link shall reference only a stratum_technology through the related attribute.

WR4: The stratum_technology_link shall reference only a stratum_technology through the relating attribute.

WR5: The stratum_technology_link name inherited from property_definition shall be blank.

WR6: The stratum_technology_link description inherited from property_definition shall be blank.

WR7: The stratum_technology_link name inherited from property_definition_relationship shall be blank.

WR8: The stratum_technology_link description inherited from property_definition_relationship shall be blank.

WR9: The stratum_technology_link shall be referenced exactly once by a part_template_definition that has a description of 'printed part template'.

WR10: The stratum_technology_link shall be referenced as a relating_property_definition no more than once by another stratum_technology_link.

WR11: The stratum_technology_link shall be referenced as a related_property_definition no more than once by another stratum_technology_link.

NOTE The stratum_technology_link that is not referenced as a related_property_definition but that is referenced by a relating_property_definition attribute of a property_definition_relationship with a name of 'precedent stratum technology link' is considered the initial link. The stratum_technology itself has no instance information and the sequence is only defined for printed part template purposes. The actual stackup in the interconnect is not formally constrained in this standard.

5.2.3.1.350 stratum_type_independent_padstack_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A stratum_type_independent_padstack_definition is a product_definition that implements the ARM concepts of: Stratum_type_independent_padstack_definition (see 4.2.873)

EXPRESS specification:

```
*)
ENTITY stratum_type_independent_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- stratum_type_independent_padstack_definition
(*
```

5.2.3.1.351 structured_text_representation_context

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A structured_text_representation_context is a context that implements the ARM concept of Structured_text (see 4.2.874) in association with the structured_text_representation_item.

EXPRESS specification:

```
*)
ENTITY structured_text_representation_context
  SUBTYPE OF (representation_context);
WHERE

  WR1: SIZEOF (QUERY (rep <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.CONTEXT_OF_ITEMS') | NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRUCTURED_TEXT_REPRESENTATION_ITEM' IN
    TYPEOF (rep)))) = 0;
END_ENTITY; -- structured_text_representation_context
(*
```

Formal propositions:

WR1: The structured_text_representation_context shall only be a context for representations that contain structured_text_representation_items.

5.2.3.1.352 structured_text_representation_item

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A structured_text_representation_item is a representation and a descriptive_representation_item that implements the ARM concept of Structured_text (see 4.2.874).

EXPRESS specification:

```

*)
ENTITY structured_text_representation_item
  SUBTYPE OF (representation, descriptive_representation_item);
WHERE

  WR1: SIZEOF (QUERY (rr <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRUCTURED_TEXT_REPRESENTATION_ITEM' IN
    TYPEOF (rr.rep_1) ))<= 1;

  WR2: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) >= 1;

  WR3: (SIZEOF (QUERY (it <* SELF\representation.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_REPRESENTATION_ITEM' IN
    TYPEOF (it))) = 1);
END_ENTITY; -- structured_text_representation_item
( *
```

Formal propositions:

WR1: The structured_text_representation_item shall relate to at most one structured_text_representation_item through a representation_relationship.

WR2: The structured_text_representation_item shall be referenced by at least one document.

WR3: The structured_text_representation_item shall contain exactly one externally_defined_representation_item.

5.2.3.1.353 supplied_part_relationship

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `supplied_part_relationship` relates the identifications of two **products** and specifies that one is the identification used by the internal design organization and the other is the identification used by a supplier.

EXPRESS specification:

```
*)
ENTITY supplied_part_relationship
  SUBTYPE OF (product_definition_relationship);
WHERE

  WR1: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE' IN
        TYPEOF (SELF.related_product_definition.formation);

  WR2: NOT ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION' IN ROLESOF (SELF));
END_ENTITY; -- supplied_part_relationship
(*)
```

Formal propositions:

WR1: The `supplied_part_relationship` shall have a `related_product_definition` in which the source is specified.

WR2: The `supplied_part_relationship` shall not be the definition of a `product_definition_shape`.

5.2.3.1.354 **surface_prepped_terminal**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `surface_prepped_terminal` is a `package_terminal` that implements the ARM concept of `Surface_prepped_terminal` (see 4.2.881).

EXPRESS specification:

```
*)
ENTITY surface_prepped_terminal
  SUBTYPE OF (altered_package_terminal);
WHERE
```

```

WR2: (SIZEOF (QUERY (ada <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION' IN
    TYPEOF (ada.assigned_document)) AND
    (ada.assigned_document.kind\document_type.product_data_type =
    'surface finish specification')) = 1);
END_ENTITY; -- surface_prepped_terminal
(*

```

Formal propositions:

WR1: The `surface_prepped_terminal` shall have exactly one `document_reference` where the `assigned_document` is an `ee_specification` with a `product_data_type` of 'surface finish specification'.

5.2.3.1.355 **surface_profile_tolerance**

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `surface_profile_tolerance` is a `physical_unit_geometric_tolerance` that implements the ARM concepts of:

- `Composite_surface_profile_locating_tolerance` (see 4.2.165);
- `Composite_surface_profile_refinement_tolerance` (see 4.2.166);
- `Profile_of_any_surface_tolerance` (see 4.2.747);
- `Profile_of_any_surface_tolerance_with_referenced_datum_system` (see 4.2.748);
- `Profile_of_any_surface_tolerance_without_referenced_datum_system` (see 4.2.749).

EXPRESS specification:

```

*)
ENTITY surface_profile_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
WHERE

```



```

WR1: (NOT(SELF\geometric_tolerance.name =
        'surface profile refinement'))
OR (SIZEOF(QUERY(gtr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.relatng_geometric_tolerance))
AND ((gtr.relatng_geometric_tolerance\geometric_tolerance.name =
    'surface profile locating')
OR (gtr.relatng_geometric_tolerance\geometric_tolerance.name =
    'surface profile refinement'))
AND (gtr\geometric_tolerance_relationship.name = 'surface profile refining control'
)) = 1);

WR2: (NOT(SELF\geometric_tolerance.name = 'surface profile locating'))
OR (SIZEOF(QUERY(gtr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.related_geometric_tolerance))
AND (gtr.related_geometric_tolerance\geometric_tolerance.name =
    'surface profile refinement')
AND (gtr\geometric_tolerance_relationship.name = 'surface profile refining control'
)) = 1);

WR3: (NOT((NOT(
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM' IN TYPEOF(SELF)))
AND(SELF\geometric_tolerance.name = 'surface profile refinement'))
OR (SIZEOF(QUERY(gtr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE' IN
    TYPEOF(gtr.related_geometric_tolerance))
AND (gtr.related_geometric_tolerance\geometric_tolerance.name =
    'surface profile refinement')
AND (gtr\geometric_tolerance_relationship.name = 'surface profile refining control'
)) = 0);

```

```

WR4: (NOT(SELF\geometric_tolerance.name =
        'surface profile refinement'))
OR (SIZEOF(QUERY(gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE') |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SURFACE_PROFILE_TOLERANCE' IN
        TYPEOF(gtr.related_geometric_tolerance))
AND (gtr.related_geometric_tolerance\geometric_tolerance.name =
        'surface profile refinement')
AND (gtr\geometric_tolerance_relationship.name = 'surface profile refining control'
)) = 1);

WR5: NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF));
END_ENTITY; -- surface_profile_tolerance
(*)

```

Formal propositions:

WR1: If the surface_profile_tolerance has a name of 'surface profile refinement', it shall be related as the 'surface profile refining control' to exactly one surface_profile_tolerance with a name of either 'surface profile locating' or 'surface profile refinement'.

WR2: If the surface_profile_tolerance has a name of 'surface profile locating', it shall relate to exactly one surface_profile_tolerance with a name of 'surface profile refinement' to specify the 'surface profile refining control'.

WR3: If the surface_profile_tolerance is not a geometric_tolerance_with_specified_datum_system and has a name of 'surface profile refinement', it shall not relate to a surface_profile_tolerance with a name of 'surface profile refinement' to specify the 'surface profile refining control'.

WR4: If the surface_profile_tolerance has a name of 'surface profile refinement', it shall relate to at most one surface_profile_tolerance with a name of 'surface profile refinement' to specify the 'surface profile refining control'.

WR5: The surface_profile_tolerance shall not be a modified_geometric_tolerance.

Informal propositions:

IP1: If surface_profile_tolerance is a geometric_tolerance_with_specified_datum_system, it shall have a name of either 'surface profile', 'surface profile refinement', or 'surface profile locating'; otherwise, it shall have the name of either 'surface profile' or 'surface profile refinement'.

IP2: If the surface_profile_tolerance has a name of 'surface profile refinement', it shall be related as the 'surface profile refining control' to exactly one surface_profile_tolerance with a name of either 'surface profile locating' or 'surface profile refinement'. Both of these surface_profile_tolerances shall specify the same shape_aspect as the toleranced_shape_aspect.

IP3: If the `surface_profile_tolerance` has a name of 'surface profile refinement' and it is related as the 'surface profile refining control' to exactly one `surface_profile_tolerance` with a name of 'surface profile refinement', it shall either not reference a `datum_system` or it shall have less `physical_unit_datum` in its referenced `datum_system` than the `datum_system` referenced by the `surface_profile_tolerance` to which it is related as the 'surface profile refining control'.

IP4: If the `surface_profile_tolerance` has a name of 'surface profile refinement' and it is related as the 'surface profile refining control' to exactly one `surface_profile_tolerance` with a name of 'surface profile locating', it shall either not reference a `datum_system` or it shall have less than or an equal number of `physical_unit_datum` in its referenced `datum_system` than the `datum_system` referenced by the `surface_profile_tolerance` to which it is related as the 'surface profile refining control'.

IP5: If the `surface_profile_tolerance` has a name of 'surface profile refinement' and it is a `geometric_tolerance_with_specified_datum_system`, the `physical_unit_datum` of its referenced `datum_system` shall be `physical_unit_datum` of the `datum_system` of the `surface_profile_tolerance` to which it is related as the 'surface profile refining control' and shall be assigned the same precedence.

IP6: If the `surface_profile_tolerance` has a name of 'surface profile refinement' and it is a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition', the `surface_profile_tolerance` to which it is related as the 'surface profile refining control' shall also be a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property without material conditions' to specify its 'referenced datum system definition'.

IP7: If the `surface_profile_tolerance` has a name of 'surface profile refinement' and it is a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', the `surface_profile_tolerance` to which it is related as the 'surface profile refining control' shall also be a `geometric_tolerance_with_specified_datum_system` that relates to a `property_definition` of a `datum_system` that has the description of 'datum system property with material conditions' to specify its 'referenced datum system definition', and each of these `property_definition` shall relate to `property_definition` with a description of 'datum feature material condition property' to specify their 'applied material condition property'. These `property_definition` shall be applied to `physical_unit_datum_features` in the context of the `datum_systems` referenced by each of the `surface_profile_tolerance`. The name of the `property_definition` applied to each of the `physical_unit_datum_feature` in the context of the `datum_system` referenced by this `surface_profile_tolerance` shall be the same as the name of the `property_definition` applied to the same `physical_unit_datum_feature` in the context of the `datum_system` referenced by the `surface_profile_tolerance` to which this `surface_profile_tolerance` is related as the 'surface profile refining control'.

IP8: If the `surface_profile_tolerance` has a name of 'surface profile refinement', it shall specify a `measure_with_unit` as its magnitude that is less than the `measure_with_unit` specified as the magnitude of the `surface_profile_tolerance` to which it is related as the 'surface profile refining control'.

IP9: If the `surface_profile_tolerance` has a name of 'surface profile refinement', there shall exist a `property_definition` that has a description of 'true profile' that is related as the 'basic profile' to a `property_definition` of the `tolerance_zone_boundary` of this `surface_profile_tolerance` and to a `property_definition` of the `tolerance_zone_boundary` of the `surface_profile_tolerance` that this `surface_profile_tolerance` is related to as the 'surface profile refining control', and these `property_definition` shall have the description of 'boundary zone definition with specified size' and a name of either 'profile boundary' or 'profile boundary with offsets'.

IP10: The `toleranced_shape_aspect` of the `surface_profile_tolerance` shall either be a `part_template_definition`, lie on the boundary of the shape of the product, or be a `composite_shape_aspect` the 'composing' `shape_aspects` of which are all of type `part_template_definition`.

IP11: The `tolerance_zone_boundary` of the `surface_profile_tolerance` shall have a `property_definition` with a description of 'boundary zone definition with specified size' and a name of either 'profile boundary' or 'profile boundary with offsets'.

5.2.3.1.356 symmetry_tolerance

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `symmetry_tolerance` is a `geometric_tolerance_with_specified_datum_system` that implements the ARM concept of `Symmetry_tolerance` (see 4.2.883).

EXPRESS specification:

```
* )
ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'symmetry';
END_ENTITY; -- symmetry_tolerance
( *
```

Formal propositions:

WR1: The name for the `symmetry_tolerance` shall be 'symmetry'.

Informal propositions:

IP1: The `toleranced_shape_aspect` of the `symmetry_tolerance` shall either be a `centre_of_symmetry` with a description of 'plane', or a `composite_shape_aspect` in which the 'composing' `shape_aspects` are all of type `centre_of_symmetry` with a description of 'plane'.

IP2: The symmetry_tolerance shall have at most two physical_unit_datum in its referenced datum_system.

IP3: The symmetry_tolerance shall have at least one physical_unit_datum in its referenced datum_system that has description of 'plane'.

IP4: The tolerance_zone_boundary of the symmetry_tolerance shall be either a tolerance_zone_explicit_opposing_boundary_set or a tolerance_zone_implicit_opposing_boundary_set.

5.2.3.1.357 table_representation_item

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A table_representation_item is a compound_representation_item that implements the ARM concept of Characteristic_data_table (see 4.2.100).

EXPRESS specification:

```
*)
ENTITY table_representation_item
  SUBTYPE OF (compound_representation_item);
WHERE

  WR1: SIZEOF (USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS')) > 0;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIST_REPRESENTATION_ITEM'
    IN TYPEOF(SELF\compound_representation_item.item_element);
END_ENTITY; -- table_representation_item
(*
```

Formal propositions:

WR1: The table_representation_item shall be referenced by exactly one applied_document_reference.

WR2: The item_element attribute shall be a list_representation_item.

5.2.3.1.358 template_material_cross_section_boundary

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A template_material_cross_section_boundary is a shape_aspect that implements the ARM concepts of: Template_material_cross_section_boundary (see 4.2.888)

EXPRESS specification:

```

*)
ENTITY template_material_cross_section_boundary
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: SIZEOF(TYPEOF(SELF) - TYPEOF(SELF\shape_aspect ||
    SELF\template_material_cross_section_boundary)) = 0;

  WR2: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_MATERIAL' IN TYPEOF
    (SELF.of_shape\property_definition.definition);

  WR3: SELF\shape_aspect.description IN
    ['top', 'left', 'right', 'bottom'];
END_ENTITY; -- template_material_cross_section_boundary
(

```

Formal propositions:

WR1: The template_material_cross_section_boundary shall not participate in a complex instance.

WR2: The template_material_cross_section_boundary shall be an aspect of a printed_part_template_material.

WR3: The land_physical_template description shall only be 'top', 'bottom', 'left' or 'right'.

5.2.3.1.359 template_stratum_assignment

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A template_stratum_assignment relates a specific part_template_definition to a specific stratum for the purposes of binding them for one or more padstack definition.

EXPRESS specification:

```

*)
ENTITY template_stratum_assignment
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- template_stratum_assignment
(

```

5.2.3.1.360 tertiary_orientation_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A tertiary_orientation_feature is a physical_unit_datum_feature that implements the ARM concept of Tertiary_orientation_feature (see 4.2.893).

EXPRESS specification:

```

*)
ENTITY tertiary_orientation_feature
  SUBTYPE OF (physical_unit_datum_feature);
WHERE

  WR1: SIZEOF ((TYPEOF (SELF)) *
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRIMARY_ORIENTATION_FEATURE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SECONDARY_ORIENTATION_FEATURE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VIEWING_PLANE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT']) = 0;

  WR2: SELF.product_definitional = TRUE;

  WR3: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_BOTTOM_SURFACE' IN
    TYPEOF (it.relying_shape_aspect))
    )) <= 1;

  WR4: SIZEOF (QUERY (it <* QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    sar\shape_aspect_relationship.name = 'associated body vertical extent') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_BODY_TOP_SURFACE' IN
    TYPEOF (it.relying_shape_aspect))
    )) <= 1;
END_ENTITY; -- tertiary_orientation_feature
(

```

Formal propositions:

WR1: The tertiary_orientation_feature shall not be a primary_orientation_feature, a secondary_orientation_feature, a seating_plane, a viewing_plane, a component_shape_aspect, or a physical_unit_datum.

WR2: A tertiary_orientation_feature shall be on the surface of the product.

WR3: A tertiary_orientation_feature shall be associated with no more than one package_body_bottom_surface.

WR4: A tertiary_orientation_feature shall be associated with no more than one package_body_top_surface.

5.2.3.1.361 test_point_part_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A test_point_part_feature is a part_tooling_feature that implements the ARM concept of Fiducial_part_feature (see 4.2.374).

EXPRESS specification:

```
* )
ENTITY test_point_part_feature
  SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- test_point_part_feature
( *
```

5.2.3.1.362 thermal_component

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_component is a component_definition that implements the ARM concept of Thermal_component (see 4.2.900).

EXPRESS specification:

```
* )
ENTITY thermal_component
  SUBTYPE OF (component_definition);
END_ENTITY; -- thermal_component
( *
```

5.2.3.1.363 thermal_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_component_shape_aspect is a component_shape_aspect that implements the ARM concept of Thermal_component (see 4.2.900).

EXPRESS specification:

```
*)
ENTITY thermal_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
END_ENTITY; -- thermal_component_shape_aspect
(*
```

5.2.3.1.364 thermal_feature

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_feature is a shape_aspect that implements the ARM concept of Thermal_feature (see 4.2.901).

EXPRESS specification:

```
*)
ENTITY thermal_feature
  SUBTYPE OF (shape_aspect);
END_ENTITY; -- thermal_feature
(*
```

5.2.3.1.365 thermal_isolation_component_shape_aspect

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_isolation_component_shape_aspect is a filled_area_material_removal_component_shape_aspect that implements the ARM concept of Thermal_isolation_laminate_component (see 4.2.902).

EXPRESS specification:

```
*)
ENTITY thermal_isolation_component_shape_aspect
  SUBTYPE OF (filled_area_material_removal_component_shape_aspect);
END_ENTITY; -- thermal_isolation_component_shape_aspect
(*
```

5.2.3.1.366 thermal_isolation_removal_template_definition

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_isolation_removal_template_definition is a part_template_definition that implements the ARM concept of Thermal_isolation_removal_template (see 4.2.904) .

EXPRESS specification:

```

*)
ENTITY thermal_isolation_removal_template_definition
  SUBTYPE OF (part_template_definition);
( *
WHERE

  WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF (QUERY (it <* pdr.used_representation.items |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TEXT_LITERAL' IN
    TYPEOF (it))) = 1)) = 1))) = 0));

  WR2: (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    SIZEOF (QUERY (it <* pdr.used_representation.items |
    (SIZEOF (
    ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND (it\representation_item.name =
    'maximum font vertical extent')) = 1
    )) = 1))) = 0));

```

```

WR3: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  SIZEOF (QUERY (it <* pdr.used_representation.items |
  (SIZEOF (
  ['ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM',
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2) AND (it\representation_item.name =
    'maximum font horizontal extent')))) = 1
  )) = 1))) = 0);
*)
END_ENTITY; -- thermal_isolation_removal_template_definition
(*

```

5.2.3.1.367 thermal_network

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A thermal_network is a functional_unit that implements the ARM concept of Thermal_network_definition (see 4.2.906).

EXPRESS specification:

```

*)
ENTITY thermal_network
  SUBTYPE OF (functional_unit);
END_ENTITY; -- thermal_network
(*

```

5.2.3.1.368 tolerance_zone_boundary

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A tolerance_zone_boundary is a shape_aspect that implements the ARM concepts of:

- Conical_tolerance_zone_boundary (see 4.2.178);
- Tolerance_zone_boundary (see 4.2.915);
- Tolerance_zone_circular_or_cylindrical_or_spherical_boundary (see 4.2.919).

EXPRESS specification:

```

*)
ENTITY tolerance_zone_boundary
  SUBTYPE OF (shape_aspect);
WHERE

  WR1: (NOT(SELF\shape_aspect.description = 'conical'))
    OR (NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET'
      IN TYPEOF(SELF))
    OR('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET'
      IN TYPEOF(SELF))));

  WR2: (NOT(SELF\shape_aspect.description =
    'circular or cylindrical or spherical'))
    OR (NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET'
      IN TYPEOF(SELF))
    OR('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET'
      IN TYPEOF(SELF))));

  WR3: SELF\shape_aspect.product_definitional = False;

  WR4: NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET'
    IN TYPEOF(SELF))
    AND ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET'
    IN TYPEOF(SELF)));

  WR5: SIZEOF(QUERY(pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (pd\property_definition.description =
    'boundary zone definition with specified size') OR
    (pd\property_definition.description = 'conical tolerance zone boundary')
    )) = 1;

  WR6: SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TOLERANCE_ZONE_DEFINITION.BOUNDARIES')) = 1;
END_ENTITY; -- tolerance_zone_boundary
(

```

Formal propositions:

WR1: If the description of the tolerance_zone_boundary is 'conical', it shall not be either a tolerance_zone_explicit_opposing_boundary_set or a tolerance_zone_implicit_opposing_boundary_set.

WR2: If the description of the tolerance_zone_boundary is 'circular or cylindrical or spherical', it shall not be either a tolerance_zone_explicit_opposing_boundary_set or a tolerance_zone_implicit_opposing_boundary_set.

WR3: The tolerance_zone_boundary shall not lie on the boundary of the shape of the product.

WR4: The tolerance_zone_boundary shall not be both a tolerance_zone_explicit_opposing_boundary_set and a tolerance_zone_implicit_opposing_boundary_set.

WR5: The tolerance_zone_boundary shall have exactly one property_definition that has the description of either 'boundary zone definition with specified size' or 'conical tolerance zone boundary'.

WR6: The tolerance_zone_boundary shall be specified as the boundaries by exactly one tolerance_zone_definition.

Informal propositions:

IP1: If the description of the tolerance_zone_boundary is 'conical', the tolerance_zone_boundary shall relate to exactly two shape_aspect to specify the 'conical tolerance zone boundary and surface'.

IP2: If the description of the tolerance_zone_boundary is 'circular or cylindrical or spherical', the tolerance_zone_boundary shall have exactly one property_definition with a description of 'boundary zone definition with specified size', and that property_definition shall be represented by exactly one representation that has a name of 'diametrical boundary size'.

5.2.3.1.369 tolerance_zone_explicit_opposing_boundary_set

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A tolerance_zone_explicit_opposing_boundary_set is a tolerance_zone_boundary that implements the ARM concepts of Tolerance_zone_explicit_opposing_boundary_set (see 4.2.923) and Tolerance_zone_opposing_boundary_set (see 4.2.926).

EXPRESS specification:

```
* )
ENTITY tolerance_zone_explicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_boundary);
END_ENTITY; -- tolerance_zone_explicit_opposing_boundary_set
(*
```

Informal propositions:

IP1: The `tolerance_zone_explicit_opposing_boundary_set` shall relate to exactly two `shape_aspect` instances that have a description of 'tolerance zone boundary member' to specify the 'zone boundary member'.

IP2: The `tolerance_zone_explicit_opposing_boundary_set` shall have exactly one `property_definition` with a description of 'boundary zone definition with specified size', and that `property_definition` shall be represented by exactly one representation that has a name of 'opposing boundary set size'.

5.2.3.1.370 `tolerance_zone_implicit_opposing_boundary_set`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `tolerance_zone_implicit_opposing_boundary_set` is a `tolerance_zone_boundary` that implements the ARM concept of `Tolerance_zone_implicit_opposing_boundary_set` (see 4.2.925).

EXPRESS specification:

```
*)
ENTITY tolerance_zone_implicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_boundary);
END_ENTITY; -- tolerance_zone_implicit_opposing_boundary_set
(*
```

Informal propositions:

IP1: The `tolerance_zone_implicit_opposing_boundary_set` shall have exactly one `property_definition` with a description of 'boundary zone definition with specified size', and that `property_definition` shall be represented by exactly one representation that has a name of 'opposing boundary set size'.

5.2.3.1.371 `total_runout_tolerance`

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A `total_runout_tolerance` is a `geometric_tolerance_with_specified_datum_system` that implements the ARM concept of `Total_runout_tolerance` (see 4.2.929).

EXPRESS specification:

```

*)
ENTITY total_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
WHERE

  WR1: SELF\geometric_tolerance.name = 'total runout';

  WR2: NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MODIFIED_GEOMETRIC_TOLERANCE' IN TYPEOF(SELF));
END_ENTITY; -- total_runout_tolerance
( *

```

Formal propositions:

WR1: The name for the total_runout_tolerance shall be 'total runout'.

WR2: The total_runout_tolerance shall not be a modified_geometric_tolerance.

Informal propositions:

IP1: The toleranced_shape_aspect of the total_runout_tolerance shall either be a part_template_definition, lie on the boundary of the shape of the product, or be a composite_shape_aspect the 'composing' shape_aspects of which are all of type part_template_definition.

IP2: The total_runout_tolerance shall have at most two physical_unit_datum instances in its referenced datum_system.

IP3: The total_runout_tolerance shall have at least one physical_unit_datum in its referenced datum_system that has a description of 'axis'.

IP4: The total_runout_tolerance shall relate to exactly one property_definition that has the description of 'datum system property without material conditions' to specify the 'referenced datum system definition'.

IP5: The tolerance_zone_boundary of the total_runout_tolerance shall be either a tolerance_zone_explicit_opposing_boundary_set or a tolerance_zone_implicit_opposing_boundary_set.

5.2.3.1.372 usage_view_connection_zone_terminal_shape_relationship

A usage_view_connection_zone_terminal_shape_relationship is a representation and a representation_relationship_with_transformation that implements the ARM concept of Usage_view_connection_zone_terminal_shape_relationship (see 4.2.943).

EXPRESS specification:

```

*)
ENTITY usage_view_connection_zone_terminal_shape_relationship
  SUBTYPE OF (representation, representation_relationship_with_transformation);
UNIQUE
  UR1: SELF\representation_relationship.rep_1, SELF\representation_relationship.rep_2;
WHERE

  WR1: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' ] *
    TYPEOF (SELF.rep_1)) = 1;

  WR2: SIZEOF ([ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION' ] *
    TYPEOF (SELF.rep_2)) = 1;

  WR3: SIZEOF (TYPEOF (SELF) - (TYPEOF (SELF\representation ||
    SELF\representation_relationship ||
    SELF\representation_relationship_with_transformation ||
    SELF\usage_view_connection_zone_terminal_shape_relationship))) = 0;

  WR4: SELF\representation_relationship.rep_1 <>
    SELF\representation_relationship.rep_2;
END_ENTITY; -- usage_view_connection_zone_terminal_shape_relationship
(*)

```

Formal propositions:

UR1: The combination of rep_1 and rep_2 shall be unique within a population of usage_view_connection_zone_terminal_shape_relationship.

WR1: The usage_view_connection_zone_terminal_shape_relationship shall reference a shape_representation as its rep_1.

WR2: The usage_view_connection_zone_terminal_shape_relationship shall reference a shape_representation as its rep_2.

WR3: The usage_view_connection_zone_terminal_shape_relationship shall only be a representation, representation_relationship, or representation_relationship_with_transformation in addition to itself.

WR4: The usage_view_connection_zone_terminal_shape_relationship rep_1 shall not be the rep_2.

5.2.3.1.373 viewing_plane

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

The viewing_plane is a shape_aspect that implements the ARM concept of Viewing_plane (see 4.2.956).

EXPRESS specification:

```

*)
ENTITY viewing_plane
  SUBTYPE OF (shape_aspect);
  WHERE

    WR1: SELF\shape_aspect.description = 'affected plane';

    WR2: (SIZEOF(QUERY ( pd <* USEDIN (SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.' + 'DEFINITION') | (pd\property_definition.description =
      'viewing plane property')))) =1;

    WR3: NOT SELF\shape_aspect.product_definitional;
END_ENTITY; -- viewing_plane
(

```

Formal propositions:

WR1: The viewing_plane shall have a description of 'affected plane'.

WR2: The viewing_plane shall have exactly one property_definition that has the description of 'viewing plane property'.

WR3: The viewing_plane shall not lie on the boundary of the shape of the product.

5.2.3.1.374 volume_shape_intersection

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A volume_shape_intersection is a representation and a shape_representation_relationship that implements the ARM concept of Volume_shape_intersection (see 4.2.960).

EXPRESS specification:

*)

ENTITY volume_shape_intersection

SUPERTYPE OF (ONEOF(edge_segment_cross_section,
pre_defined_parallel_datum_axis_symbol_3d_2d_relationship,
pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship,
pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship))

SUBTYPE OF (representation,
shape_representation_relationship,
representation_relationship_with_transformation);

UNIQUE

UR1: SELF\representation_relationship.rep_1, SELF\representation_relationship.rep_2;

WHERE

WR1: SELF\representation_relationship.rep_1 :<>: SELF\representation_relationship.rep_2;

WR2: (SIZEOF(TYPEOF(SELF) - (TYPEOF(SELF\representation ||
SELF\representation_relationship ||
SELF\shape_representation_relationship ||
SELF\representation_relationship_with_transformation ||
SELF\volume_shape_intersection))) = 0) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_SEGMENT_CROSS_SECTION' IN TYPEOF (SELF)) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRE_DEFINED_PARALLEL_DATUM_AXIS_SYMBOL_3D_2D_RELATIONSHIP' IN TYPEOF (SELF)) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRE_DEFINED_PERPENDICULAR_DATUM_AXIS_SYMBOL_3D_2D_RELATIONSHIP' IN TYPEOF (SELF)) OR
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRE_DEFINED_PERPENDICULAR_DATUM_PLANE_SYMBOL_3D_2D_RELATIONSHIP' IN TYPEOF (SELF))

WR3: 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ITEM_DEFINED_TRANSFORMATION' IN TYPEOF
(SELF\representation_relationship_with_transformation.transformation_operator);

END_ENTITY; -- volume_shape_intersection

(*

Formal propositions:

UR1: The combination of rep_1 and rep_2 shall be unique within a population of volume_shape_intersection.

WR1: The volume_shape_intersection rep_1 shall not be rep_2.

WR2: The volume_shape_intersection shall not be a complex subtype of other than its direct supertypes.

WR3: The volume_shape_intersection shall reference an item_defined_transformation.

5.2.3.1.375 wire_terminal

NOTE The local rules for this entity have not been fully updated to agree with the changes in clause 4.2.

A wire_terminal is a package_terminal that implements the ARM concept of Wire_terminal (see 4.2.963).

EXPRESS specification:

```

*)
ENTITY wire_terminal
  SUBTYPE OF (package_terminal);
WHERE

WR1: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF (QUERY (it <* pdr.used_representation.items |
  (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM') IN TYPEOF (it))
  AND
  (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_UNIT') IN
  TYPEOF (it\measure_with_unit.unit_component)))) = 2))) = 1)) )) = 0);

WR2: (SIZEOF (QUERY (pd <* USEDIN (SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF (QUERY (it <* pdr.used_representation.items | (
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
  (it\representation_item.name = 'maximum wire terminal length')
  AND
  (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LENGTH_UNIT') IN
  TYPEOF (it\measure_with_unit.unit_component)))) = 1
  ))) = 1)) )) = 0);

```

```

WR3: (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    (NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF (QUERY (it <* pdr.used_representation.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
    (it\representation_item.name = 'minimum wire terminal length')
    AND
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_UNIT') IN
    TYPEOF (it\measure_with_unit.unit_component)))) = 1)))
    = 1)) )) = 0);

WR4: (SIZEOF (QUERY (sar <* USEDIN (SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name = 'internal connection zone') AND
    (sar.related_shape_aspect\shape_aspect.description = 'connection zone')) = 1);
END_ENTITY; -- wire_terminal
(*

```

Formal propositions:

WR1: The wire_terminal shall have a representation that contains exactly two length_measure_with_-units.

WR2: The wire_terminal shall have a representation that contains exactly one length_measure_with_-unit with a name of 'maximum wire terminal length'.

WR3: The wire_terminal shall have a representation that contains exactly one length_measure_with_-unit with a name of 'minimum wire terminal length'.

WR4: The wire_terminal shall be related to exactly one shape_aspect with a description of 'connection zone' as the 'internal connection zone'.

5.2.3.2 electronic_assembly_interconnect_and_packaging_design imported entity modifications

5.2.3.2.1 action

The base definition of the action entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of action is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the action entity:

- rule_supersedence_constraint (See ??) ;
- subtype_mandatory_action (See ??) .

5.2.3.2.2 action_assignment

The base definition of the action_assignment entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of action_assignment is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the action_assignment entity:

- subtype_combination_action_assignment (See ??) .

5.2.3.2.3 action_directive

The base definition of the action_directive entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of action_directive is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the action_directive entity:

- dependent_instantiable_action_directive (See 5.2.3.99) .

5.2.3.2.4 action_request_assignment

The base definition of the action_request_assignment entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of action_request_assignment is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the action_request_assignment entity:

- subtype_exclusive_action_request_assignment (See ??) .

5.2.3.2.5 action_request_status

The base definition of the action_request_status entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of action_request_status is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the action_request_status entity:

- restrict_action_request_status (See ??) ;
- versioned_action_request_requires_status (See ??) .

5.2.3.2.6 address

The base definition of the address entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of address is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the address entity:

- dependent_instantiable_address (See ??) ;
- subtype_mandatory_address (See ??) .

5.2.3.2.7 alternate_product_relationship

The base definition of the alternate_product_relationship entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of alternate_product_relationship is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the `alternate_product_relationship` entity:

- `assembly_alternate_product_constraint` (See 5.2.3.16) .

5.2.3.2.8 angular_size

The base definition of the `angular_size` entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of `angular_size` is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the `angular_size` entity:

- `angular_size_dimension_constraint` (See 5.2.3.10) .

5.2.3.2.9 annotation_fill_area_occurrence

The base definition of the `annotation_fill_area_occurrence` entity is given in ISO 10303-46. The following modifications apply to this Part of ISO 10303.

The definition of `annotation_fill_area_occurrence` is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the `annotation_fill_area_occurrence` entity:

- `fill_area_occurrence_constraint` (See ??) .

5.2.3.2.10 application_context

The base definition of the `application_context` entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of `application_context` is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the `application_context` entity:

- `application_context_requires_ap_definition` (See 5.2.3.11) .

5.2.3.2.11 application_protocol_definition

The base definition of the application_protocol_definition entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of application_protocol_definition is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the application_protocol_definition entity:

- application_context_requires_ap_definition (See 5.2.3.11) .

5.2.3.2.12 approval

The base definition of the approval entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of approval is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the approval entity:

- approval_requires_approval_date_time (See 5.2.3.12) ;
- approval_requires_approval_person_organization (See 5.2.3.13) ;
- approvals_are_assigned (See 5.2.3.15) .

5.2.3.2.13 approval_assignment

The base definition of the approval_assignment entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of approval_assignment is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the approval_assignment entity:

- approvals_are_assigned (See 5.2.3.15) .

5.2.3.2.14 approval_date_time

The base definition of the approval_date_time entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of approval_date_time is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the approval_date_time entity:

- approval_requires_approval_date_time (See 5.2.3.12) .

5.2.3.2.15 approval_person_organization

The base definition of the approval_person_organization entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of approval_person_organization is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the approval_person_organization entity:

- approval_requires_approval_person_organization (See 5.2.3.13) .

5.2.3.2.16 approval_status

The base definition of the approval_status entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of approval_status is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the approval_status entity:

- dependent_instantiable_approval_status (See 5.2.3.100) ;
- restrict_approval_status (See ??) .

5.2.3.2.17 assembly_component_usage

The base definition of the assembly_component_usage entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of assembly_component_usage is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the assembly_component_usage entity:

- acu_requires_security_classification (See ??) ;
- assembly_composition_relationship_constraint (See ??) ;
- assembly_composition_relationship_unique_constraint (See 5.2.3.17) ;
- interconnect_module_stratum_assembly_relationship_constraint (See ??);
- interconnect_module_stratum_assembly_relationship_unique_constraint (See ??);
- material_assembly_relationship_constraint (See ??);
- material_assembly_relationship_unique_constraint (See ??).

5.2.3.2.18 assembly_component_usage_substitute

The base definition of the assembly_component_usage_substitute entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of assembly_component_usage_substitute is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the assembly_component_usage_-substitute entity:

- substitute_product_constraint (See ??) .

5.2.3.2.19 centre_of_symmetry

The base definition of the centre_of_symmetry entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of centre_of_symmetry is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the centre_of_symmetry entity:

- centre_of_symmetry_constraint (See 5.2.3.28) .

5.2.3.2.20 certification_type

The base definition of the certification_type entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of certification_type is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the certification_type entity:

- dependent_instantiable_certification_type (See 5.2.3.101) .

5.2.3.2.21 characterized_object

The base definition of the characterized_object entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of characterized_object is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the characterized_object entity:

- dependent_instantiable_certification_type (See 5.2.3.101) ;
- subtype_mandatory_characterized_object (See ??) .

5.2.3.2.22 composite_shape_aspect

The base definition of the composite_shape_aspect entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of composite_shape_aspect is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the composite_shape_aspect entity:

- composite_shape_element_constraint (See 5.2.3.57) ;
- part_feature_unique_constraint (See ??) ;
- subtype_mandatory_composite_shape_aspect (See ??) .

5.2.3.2.23 configuration_effectivity

The base definition of the configuration_effectivity entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of configuration_effectivity is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the configuration_effectivity entity:

- planned_effectivity_constraint (See ??) ;
- planned_effectivity_unique_constraint (See ??) .

5.2.3.2.24 configuration_item

The base definition of the configuration_item entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of configuration_item is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the configuration_item entity:

- configuration_item_requires_person_organization (See 5.2.3.63) ;
- ee_product_configuration_constraint (See ??) ;
- ee_product_configuration_unique_constraint (See ??) ;
- product_concept_requires_configuration_item (See ??).

5.2.3.2.25 contract

The base definition of the contract entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of contract is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the contract entity:

- contract_constraint (See ??) ;
- contract_unique_constraint (See 5.2.3.69) .

5.2.3.2.26 contract_type

The base definition of the contract_type entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of contract_type is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the contract_type entity:

- dependent_instantiable_contract_type (See 5.2.3.102) .

5.2.3.2.27 curve_style

The base definition of the curve_style entity is given in ISO 10303-46. The following modifications apply to this Part of ISO 10303.

The definition of curve_style is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the curve_style entity:

- curve_style_requires_length_measure_with_unit (See 5.2.3.72) .

5.2.3.2.28 date

The base definition of the date entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of date is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the date entity:

- date_constraint (See 5.2.3.75) ;
- dependent_instantiable_date (See 5.2.3.103) ;
- subtype_mandatory_date (See ??) .

5.2.3.2.29 date_and_time

The base definition of the date_and_time entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of date_and_time is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the date_and_time entity:

- date_and_time_require_minute_and_second (See 5.2.3.74) .

5.2.3.2.30 date_role

The base definition of the date_role entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of date_role is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the date_role entity:

- dependent_instantiable_date_role (See 5.2.3.104) ;
- restrict_date_role (See ??) .

5.2.3.2.31 date_time_role

The base definition of the date_time_role entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of date_time_role is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the date_time_role entity:

- dependent_instantiable_date_time_role (See 5.2.3.105) .

5.2.3.2.32 derived_shape_aspect

The base definition of the derived_shape_aspect entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of derived_shape_aspect is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the derived_shape_aspect entity:

- derived_shape_element_constraint (See ??) ;
- directed_axis_constraint (See ??) ;
- directed_axis_unique_constraint (See ??) .

5.2.3.2.33 descriptive_representation_item

The base definition of the descriptive_representation_item entity is given in ISO 10303-45. The following modifications apply to this Part of ISO 10303.

The definition of descriptive_representation_item is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the descriptive_representation_item entity:

- component_feature_external_reference_constraint (See 5.2.3.36) ;
- component_feature_external_reference_unique_constraint (See 5.2.3.37) ;
- configuration_controlled_message_constraint (See 5.2.3.62) ;
- package_terminal_external_reference_constraint (See ??);
- package_terminal_external_reference_unique_constraint (See ??).

5.2.3.2.34 dimensional_location

The base definition of the dimensional_location entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of dimensional_location is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the dimensional_location entity:

- dimensional_characteristic_constraint (See ??) ;
- dimensional_location_constraint (See ??) ;
- shape_element_locating_relationship_constraint (See ??) ;
- shape_element_locating_relationship_unique_constraint (See ??);
- subtype_exclusive_dimensional_location (See ??).

5.2.3.2.35 dimensional_size

The base definition of the dimensional_size entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of dimensional_size is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the dimensional_size entity:

- dimensional_characteristic_constraint (See ??) ;
- dimensional_size_constraint (See ??) ;
- subtype_exclusive_dimensional_size (See ??) .

5.2.3.2.36 directed_action

The base definition of the directed_action entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of directed_action is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the directed_action entity:

- directed_action_requires_approval (See ??) ;
- directed_action_requires_date_or_date_and_time (See ??) ;
- work_order_constraint (See ??) ;
- work_order_unique_constraint (See ??).

5.2.3.2.37 document

The base definition of the document entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of document is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the document entity:

- ee_document_constraint (See ??) ;
- ee_document_identifier_unique_constraint (See ??) ;
- ee_document_unique_constraint (See ??) .

5.2.3.2.38 **document_type**

The base definition of the document_type entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of document_type is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the document_type entity:

— dependent_instantiable_document_type (See ??) .

5.2.3.2.39 **externally_defined_item**

The base definition of the externally_defined_item entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of externally_defined_item is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the externally_defined_item entity:

— subtype_mandatory_externally_defined_item (See ??) .

5.2.3.2.40 **externally_defined_text_font**

The base definition of the externally_defined_text_font entity is given in ISO 10303-46. The following modifications apply to this Part of ISO 10303.

The definition of externally_defined_text_font is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the externally_defined_text_font entity:

- registered_font_constraint (See ??) ;
- registered_font_unique_constraint (See ??) .

5.2.3.2.41 geometric_tolerance

The base definition of the geometric_tolerance entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of geometric_tolerance is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the geometric_tolerance entity:

- subtype_mandatory_geometric_tolerance (See ??) .

5.2.3.2.42 global_unit_assigned_context

The base definition of the global_unit_assigned_context entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of global_unit_assigned_context is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the global_unit_assigned_context entity:

- cartesian_coordinate_system_constraint (See 5.2.3.26) .

5.2.3.2.43 group

The base definition of the group entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of group is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the group entity:

- subtype_mandatory_group (See ??) .

5.2.3.2.44 make_from_usage_option

The base definition of the make_from_usage_option entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of make_from_usage_option is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the make_from_usage_option entity:

- no_shape_for_make_from (See ??) ;
- stratum_make_from_relationship_constraint (See ??) .

5.2.3.2.45 mapped_item

The base definition of the mapped_item entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of mapped_item is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the mapped_item entity:

- design_intent_modification_in_stratum_3d_position_constraint (See ??) ;
- design_intent_modification_in_stratum_planar_position_constraint (See ??) ;
- device_2d_position_constraint (See ??) ;
- device_3d_position_constraint (See ??);
- stratum_feature_in_stratum_3d_position_constraint (See ??);
- stratum_feature_in_stratum_planar_position_constraint (See ??);
- stratum_interconnect_module_3d_position_constraint (See ??);
- stratum_interconnect_module_planar_position_constraint (See ??);
- stratum_surface_in_stratum_3d_position_constraint (See ??);
- subtype_exclusive_mapped_item (See ??).

5.2.3.2.46 material_designation

The base definition of the material_designation entity is given in ISO 10303-45. The following modifications apply to this Part of ISO 10303.

The definition of material_designation is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the material_designation entity:

- ee_material_constraint (See ??) ;
- ee_material_unique_constraint (See ??) ;
- printed_part_template_material_constraint (See ??) .

5.2.3.2.47 measure_qualification

The base definition of the measure_qualification entity is given in ISO 10303-45. The following modifications apply to this Part of ISO 10303.

The definition of measure_qualification is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the measure_qualification entity:

- dimension_qualifier_constraint (See ??) ;
- geometric_tolerance_qualifier_constraint (See ??) ;
- pre_defined_dimension_qualifier_constraint (See ??) ;
- pre_defined_geometric_tolerance_qualifier_constraint (See ??);
- user_defined_dimension_qualifier_constraint (See ??);
- user_defined_geometric_tolerance_qualifier_constraint (See ??).

5.2.3.2.48 modified_geometric_tolerance

The base definition of the modified_geometric_tolerance entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of modified_geometric_tolerance is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the modified_geometric_tolerance entity:

- geometric_tolerance_with_applied_material_condition_property_constraint (See ??) .

5.2.3.2.49 **named_unit**

The base definition of the `named_unit` entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of `named_unit` is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the `named_unit` entity:

- `dependent_instantiable_named_unit` (See ??) .

5.2.3.2.50 **next_assembly_usage_occurrence**

The base definition of the `next_assembly_usage_occurrence` entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of `next_assembly_usage_occurrence` is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the `next_assembly_usage_occurrence` entity:

- `coordinated_assembly_and_shape` (See ??) ;
- `next_higher_assembly_relationship_constraint` (See ??) ;
- `next_higher_assembly_relationship_unique_constraint` (See ??) .

5.2.3.2.51 **organization_role**

The base definition of the `organization_role` entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of `organization_role` is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the organization_role entity:

- dependent_instantiable_organization_role (See ??) .

5.2.3.2.52 parametric_representation_context

The base definition of the parametric_representation_context entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of parametric_representation_context is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the parametric_representation_context entity:

- dependent_instantiable_parametric_representation_context (See ??) .

5.2.3.2.53 person_and_organization_role

The base definition of the person_and_organization_role entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of person_and_organization_role is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the person_and_organization_role entity:

- dependent_instantiable_person_and_organization_role (See ??) .

5.2.3.2.54 plus_minus_tolerance

The base definition of the plus_minus_tolerance entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of plus_minus_tolerance is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the plus_minus_tolerance entity:

- plus_minus_tolerance_range_representation_constraint (See ??) .

5.2.3.2.55 pre_defined_item

The base definition of the pre_defined_item entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of pre_defined_item is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the pre_defined_item entity:

- subtype_mandatory_pre_defined_item (See ??) .

5.2.3.2.56 product

The base definition of the product entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product entity:

- ee_product_constraint (See ??) ;
- ee_product_unique_constraint (See ??) ;
- mandatory_product_related_product_category_value (See ??) ;
- product_requires_person_organization (See ??);
- product_requires_product_definition_formation (See ??);
- substitute_product_constraint (See ??).

5.2.3.2.57 product_concept

The base definition of the product_concept entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of product_concept is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_concept entity:

- ee_product_model_unique_constraint (See ??) ;
- product_concept_requires_configuration_item (See ??) .

5.2.3.2.58 product_definition

The base definition of the product_definition entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product_definition is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition entity:

- part_feature_constraint (See ??) ;
- product_definition_requires_date_or_date_and_time (See ??) ;
- product_definition_requires_person_organization (See ??) ;
- subtype_exclusive_product_definition (See ??).

5.2.3.2.59 product_definition_formation

The base definition of the product_definition_formation entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product_definition_formation is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition_formation entity:

- bus_structural_definition_unique_constraint (See 5.2.3.25) ;
- ee_product_version_constraint (See ??) ;
- ee_product_version_unique_constraint (See ??) ;
- pdf_requires_security_classification (See ??);
- product_definition_formation_requires_approval (See ??);
- product_definition_formation_requires_person_organization (See ??);
- product_requires_product_definition_formation (See ??).

5.2.3.2.60 product_definition_formation_relationship

The base definition of the product_definition_formation_relationship entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product_definition_formation_relationship is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the product_definition_formation_relationship entity:

- product_association_constraint (See ??) .

5.2.3.2.61 product_definition_relationship

The base definition of the product_definition_relationship entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product_definition_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the product_definition_relationship entity:

- aggregate_connectivity_requirement_unique_constraint (See 5.2.3.6) ;
- assembly_requirement_allocation_constraint (See 5.2.3.19) ;
- bus_element_relationship_constraint (See ??) ;
- bus_element_relationship_unique_constraint (See ??);
- connectivity_allocation_constraint (See ??);
- connectivity_allocation_unique_constraint (See ??);
- design_composition_path_constraint (See ??);
- design_functional_unit_allocation_to_assembly_component_constraint (See ??);
- design_functional_unit_allocation_to_assembly_component_unique_constraint (See ??);
- design_functional_unit_allocation_to_assembly_joint_constraint (See ??);
- design_functional_unit_allocation_to_assembly_joint_unique_constraint (See ??);
- design_functional_unit_allocation_to_fabrication_joint_constraint (See ??);
- design_functional_unit_allocation_to_fabrication_joint_unique_constraint (See ??);
- design_functional_unit_allocation_to_reference_functional_unit_constraint (See ??);
- design_object_management_relationship_constraint (See ??);
- equivalent_functional_unit_definition_assignment_constraint (See ??);
- functional_unit_requirement_allocation_constraint (See ??);
- interface_component_allocation_constraint (See ??);
- inter_stratum_extent_constraint (See ??);
- part_device_constraint (See ??);
- part_device_unique_constraint (See ??);

- physical_node_requirement_to_implementing_component_allocation_constraint (See ??);
- physical_unit_relationship_constraint (See ??);
- reference_composition_path_constraint (See ??);
- reference_functional_unit_assignment_to_part_constraint (See ??);
- shield_allocation_constraint (See ??);
- subtype_exclusive_product_definition_relationship (See ??);
- thermal_requirement_allocation_constraint (See ??);
- transmission_line_functional_unit_to_routed_printed_component_allocation_constraint (See ??).

5.2.3.2.62 product_definition_usage

The base definition of the product_definition_usage entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of product_definition_usage is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the product_definition_usage entity:

- subtype_mandatory_product_definition_usage (See ??) .

5.2.3.2.63 product_definition_with_associated_documents

The base definition of the product_definition_with_associated_documents entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of product_definition_with_associated_documents is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the product_definition_with_-associated_documents entity:

- ee_product_definition_constraint (See ??) .

5.2.3.2.64 product_material_composition_relationship

The base definition of the product_material_composition_relationship entity is given in ISO 10303-45. The following modifications apply to this Part of ISO 10303.

The definition of product_material_composition_relationship is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the product_material_composition_relationship entity:

- material_composition_relationship_constraint (See ??) .

5.2.3.2.65 projected_zone_definition

The base definition of the projected_zone_definition entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of projected_zone_definition is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the projected_zone_definition entity:

- projected_tolerance_zone_definition_constraint (See ??) ;
- projected_zone_and_base_relationship_constraint (See ??) ;
- projected_zone_height_characteristic_constraint (See ??) .

5.2.3.2.66 promissory_usage_occurrence

The base definition of the promissory_usage_occurrence entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of promissory_usage_occurrence is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the promissory_usage_occurrence entity:

- promissory_usage_relationship_constraint (See ??) ;
- promissory_usage_relationship_unique_constraint (See ??) .

5.2.3.2.67 property_definition

The base definition of the property_definition entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of property_definition is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_definition entity:

- boundary_zone_definition_with_specified_size_constraint (See 5.2.3.24) ;
- conical_tolerance_zone_boundary_definition_constraint (See 5.2.3.65) ;
- conical_tolerance_zone_boundary_diametrical_size_characteristic_constraint (See 5.2.3.66) ;
- datum_axis_related_orientation_constraint (See 5.2.3.77);
- datum_based_direction_vector_constraint (See 5.2.3.78);
- datum_based_vector_orientation_constraint (See 5.2.3.80);
- datum_feature_material_condition_property_constraint (See 5.2.3.81);
- datum_precedence_assignment_constraint (See 5.2.3.87);
- datum_precedence_assignment_unique_constraint (See 5.2.3.88);
- datum_system_defining_conditions_constraint (See 5.2.3.89);
- datum_system_defining_conditions_with_material_conditions_constraint (See 5.2.3.90);
- datum_system_defining_conditions_without_material_conditions_constraint (See 5.2.3.91);
- dimension_related_positional_boundary_definition_constraint (See ??);
- dimension_set_constraint (See ??);
- direction_element_constraint (See ??);
- composite_signal_property_constraint (See 5.2.3.58);
- positional_boundary_member_definition_constraint (See ??);
- positional_boundary_offset_constraint (See ??);
- profile_boundary_definition_constraint (See ??);
- profile_boundary_definition_with_offsets_constraint (See ??);
- profile_related_positional_boundary_definition_constraint (See ??);
- restraint_condition_constraint (See ??);

- signal_property_constraint (See ??);
- subtype_exclusive_property_definition (See ??);
- technology_constraint (See ??);
- tolerance_specific_restraint_condition_constraint (See ??);
- tolerance_zone_boundary_definition_constraint (See ??);
- tolerance_zone_boundary_member_offset_constraint (See ??);
- tolerance_zone_definition_with_specified_orientation_constraint (See ??);
- tolerance_zone_per_unit_size_specification_constraint (See ??);
- true_profile_constraint (See ??);
- viewing_plane_based_orientation_constraint (See ??);
- viewing_plane_definition_constraint (See ??).

5.2.3.2.68 property_definition_relationship

The base definition of the property_definition_relationship entity is given in ISO 10303-45. The following modifications apply to this Part of ISO 10303.

The definition of property_definition_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the property_definition_relationship entity:

- composite_signal_property_relationship_constraint (See ??) ;
- functional_unit_network_terminal_definition_bus_assignment_constraint (See ??) ;
- functional_unit_network_terminal_definition_node_assignment_constraint (See ??) ;
- functional_unit_terminal_bus_assignment_constraint (See ??);
- functional_unit_terminal_node_assignment_constraint (See ??);
- operational_requirement_occurrence_relationship_constraint (See ??);
- requirement_composition_constraint (See ??).

5.2.3.2.69 **property_definition_representation**

The base definition of the `property_definition_representation` entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of `property_definition_representation` is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the `property_definition_representation` entity:

- `analytical_model_vector_port_assignment_constraint` (See 5.2.3.8) ;
- `assembly_component_2d_shape_constraint` (See ??) ;
- `assembly_component_3d_shape_constraint` (See ??) ;
- `component_placement_restriction_assignment_constraint` (See 5.2.3.44);
- `functional_specification_to_requirement_occurrence_assignment_constraint` (See ??);
- `located_interconnect_module_thickness_requirement_unique_constraint` (See ??);
- `subtype_exclusive_property_definition_representation` (See ??).

5.2.3.2.70 **representation**

The base definition of the `representation` entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of `representation` is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the representation entity:

- boundary_size_characteristic_constraint (See 5.2.3.23) ;
- component_external_reference_constraint (See 5.2.3.35) ;
- component_overlap_relationship_constraint (See 5.2.3.41) ;
- component_mounting_clearance_relationship_constraint (See 5.2.3.40);
- connection_zone_constraint (See 5.2.3.68);
- curve_style_unique_constraint (See 5.2.3.73);
- datum_based_length_measure_constraint (See 5.2.3.79);
- diametrical_boundary_size_characteristic_constraint (See ??);
- ee_rule_conclusion_definition_unique_constraint (See ??);
- ee_rule_definition_constraint (See ??);
- ee_rule_definition_unique_constraint (See ??);
- ee_rule_premise_definition_unique_constraint (See ??);
- functional_specification_constraint (See ??);
- marking_constraint (See ??);
- opposing_boundary_set_size_characteristic_constraint (See ??);
- package_external_reference_constraint (See ??);
- part_terminal_external_reference_unique_constraint (See ??);
- registered_style_constraint (See ??);
- rule_priority_constraint (See ??);
- shape_element_unique_constraint (See ??);
- signal_specification_constraint (See ??);
- stratum_technology_line_width_tolerance_assignment_constraint (See ??);

- subtype_exclusive_representation (See ??).

5.2.3.2.71 representation_context

The base definition of the representation_context entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of representation_context is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the representation_context entity:

- subtype_exclusive_representation_context (See ??) .

5.2.3.2.72 representation_item

The base definition of the representation_item entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of representation_item is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the representation_item entity:

- characteristic_constraint (See 5.2.3.33) ;
- characteristic_unique_constraint (See ??) ;
- dependent_instantiable_representation_item (See ??) ;
- shape_element_unique_constraint (See ??).

5.2.3.2.73 representation_relationship

The base definition of the representation_relationship entity is given in ISO 10303-43. The following modifications apply to this Part of ISO 10303.

The definition of representation_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the representation_relationship entity:

- evaluated_characteristic_constraint (See ??) ;
- stratum_feature_shape_stratum_average_surface_shape_relationship_constraint (See ??) ;
- stratum_feature_shape_stratum_surface_shape_relationship_constraint (See ??) ;
- terminal_pair_port_equivalence_relationship_constraint (See ??).

5.2.3.2.74 runout_zone_definition

The base definition of the runout_zone_definition entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of runout_zone_definition is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the runout_zone_definition entity:

- runout_tolerance_zone_definition_with_specified_angle_constraint (See ??) .

5.2.3.2.75 runout_zone_orientation

The base definition of the runout_zone_orientation entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of runout_zone_orientation is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the runout_zone_orientation entity:

- subtype_mandatory_runout_zone_orientation (See ??) .

5.2.3.2.76 runout_zone_orientation_reference_direction

The base definition of the runout_zone_orientation_reference_direction entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of runout_zone_orientation_reference_direction is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the runout_zone_orientation_-reference_direction entity:

- runout_zone_orientation_constraint (See ??) .

5.2.3.2.77 security_classification

The base definition of the security_classification entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of security_classification is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the security_classification entity:

- security_classification_constraint (See ??) .

5.2.3.2.78 security_classification_level

The base definition of the security_classification_level entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of security_classification_level is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the security_classification_level entity:

- dependent_instantiable_security_classification_level (See ??) .

5.2.3.2.79 shape_aspect

The base definition of the shape_aspect entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of shape_aspect is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_aspect entity:

- component_termination_passage_template_terminal_constraint (See 5.2.3.55) ;
- component_termination_passage_template_terminal_unique_constraint (See 5.2.3.56) ;
- conductive_interconnect_element_with_pre_defined_transitions_constraint (See 5.2.3.60) ;
- conductive_interconnect_element_with_user_defined_single_transition_constraint (See 5.2.3.61);
- connection_zone_constraint (See 5.2.3.68);
- interconnect_module_surface_feature_constraint (See ??);
- mating_connector_termination_constraint (See ??);
- mating_connector_termination_unique_constraint (See ??);
- non_feature_shape_element_constraint (See ??);
- part_device_terminal_constraint (See ??);
- part_device_terminal_unique_constraint (See ??);
- part_feature_constraint (See ??);
- part_feature_unique_constraint (See ??);
- polarity_indication_feature_constraint (See ??);
- shape_element_constraint (See ??);
- subtype_combination_shape_aspect (See ??);
- tolerance_zone_boundary_member_constraint (See ??);
- topological_junction_constraint (See ??);
- topological_junction_unique_constraint (See ??);
- via_template_terminal_constraint (See ??);
- via_template_terminal_unique_constraint (See ??).

5.2.3.2.80 **shape_aspect_deriving_relationship**

The base definition of the shape_aspect_deriving_relationship entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of shape_aspect_deriving_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_aspect_deriving_relationship entity:

- shape_element_deriving_relationship_constraint (See ??) ;
- shape_element_deriving_relationship_unique_constraint (See ??) .

5.2.3.2.81 **shape_aspect_relationship**

The base definition of the shape_aspect_relationship entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of shape_aspect_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_aspect_relationship entity:

- adjacent_stratum_surface_definition_constraint (See 5.2.3.4) ;
- adjacent_stratum_surface_definition_unique_constraint (See 5.2.3.5) ;
- assembly_requirement_allocation_constraint (See 5.2.3.19) ;
- assembly_separation_requirement_allocation_constraint (See 5.2.3.20);
- component_feature_to_layout_feature_relationship_constraint (See 5.2.3.38);
- component_terminal_to_assembly_module_terminal_assignment_constraint (See ??);
- component_terminal_to_interconnect_module_terminal_assignment_constraint (See ??);
- component_terminal_to_interconnect_module_terminal_assignment_unique_constraint (See ??);
- component_terminal_to_pca_terminal_assignment_constraint (See 5.2.3.53);
- component_terminal_to_pca_terminal_assignment_unique_constraint (See 5.2.3.54);
- conical_tolerance_zone_boundary_and_surface_relationship_constraint (See 5.2.3.64);
- datum_axis_and_tolerance_zone_orienting_relationship_constraint (See 5.2.3.76);
- datum_feature_usage_in_common_datum_relationship_constraint (See 5.2.3.82);
- datum_feature_usage_in_datum_relationship_constraint (See 5.2.3.83);
- datum_feature_usage_in_datum_relationship_unique_constraint (See 5.2.3.84);
- datum_feature_usage_in_datum_system_relationship_constraint (See 5.2.3.85);
- datum_feature_usage_in_single_datum_relationship_constraint (See 5.2.3.86);
- datum_target_usage_in_datum_target_set_relationship_constraint (See 5.2.3.92);
- datum_target_usage_in_datum_target_set_relationship_unique_constraint (See 5.2.3.93);
- datum_usage_in_datum_system_relationship_constraint (See 5.2.3.94);
- datum_usage_in_datum_system_relationship_unique_constraint (See 5.2.3.95);

- design_intent_modification_sequence_constraint (See ??);
- design_intent_modification_sequence_unique_constraint (See ??);
- device_terminal_map_relationship_constraint (See ??);
- edge_segment_vertex_constraint (See ??);
- equivalent_functional_terminals_assignment_constraint (See ??);
- feature_group_relationship_constraint (See ??);
- functional_usage_view_to_part_terminal_assignment_constraint (See ??);
- functional_usage_view_to_part_terminal_assignment_unique_constraint (See ??);
- functional_usage_view_to_printed_part_template_terminal_assignment_constraint (See ??);
- functional_usage_view_to_printed_part_template_terminal_assignment_unique_constraint (See ??);
- internal_stratum_access_constraint (See ??);
- internal_stratum_access_unique_constraint (See ??);
- junction_vertex_allocation_constraint (See ??);
- junction_vertex_allocation_unique_constraint (See ??);
- layout_spacing_requirement_non_conforming_design_object_constraint (See ??);
- parallel_shape_element_relationship_constraint (See ??);
- perpendicular_shape_element_relationship_constraint (See ??);
- physical_connectivity_abstraction_map_unique_constraint (See ??);
- physical_connectivity_definition_relationship_constraint (See ??);
- placement_group_area_assignment_constraint (See ??);
- positional_boundary_and_profile_boundary_member_relationship_constraint (See ??);
- printed_part_template_terminal_relationship_constraint (See ??);
- runout_tolerance_zone_orienting_relationship_constraint (See ??);
- shape_element_boundary_relationship_constraint (See ??);

- shape_element_composing_relationship_constraint (See ??);
- shape_element_constituent_relationship_constraint (See ??);
- shape_element_constituent_relationship_unique_constraint (See ??);
- shape_element_relationship_constraint (See ??);
- stratum_concept_to_physical_usage_view_assignment_constraint (See ??);
- subtype_exclusive_shape_aspect_relationship (See ??);
- tolerance_zone_and_viewing_plane_orienting_relationship_constraint (See ??);
- transmission_line_functional_unit_terminal_allocation_constraint (See ??);
- transmission_line_functional_unit_terminal_allocation_unique_constraint (See ??).

5.2.3.2.82 shape_dimension_representation

The base definition of the shape_dimension_representation entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of shape_dimension_representation is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_dimension_representation entity:

- limit_dimension_representation_constraint (See ??) ;
- plus_minus_tolerance_range_representation_constraint (See ??) ;
- shape_dimension_representation_constraint (See ??) ;
- singular_dimension_representation_constraint (See ??).

5.2.3.2.83 shape_representation

The base definition of the shape_representation entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of shape_representation is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_representation entity:

- bevel_edge_feature_shape_constraint (See 5.2.3.22) ;
- default_assembly_bond_shape_constraint (See 5.2.3.96) ;
- dependent_instantiable_shape_representation (See ??) ;
- feature_shape_occurrence_constraint (See ??);
- mating_connector_placement_relationship_constraint (See ??);
- part_template_3d_shape_constraint (See ??);
- part_template_planar_shape_constraint (See ??);
- physical_unit_3d_shape_constraint (See ??);
- physical_unit_planar_shape_constraint (See ??);
- planar_curve_constraint (See ??);
- planar_projected_shape_constraint (See ??);
- radius_edge_feature_shape_constraint (See ??);
- shape_representation_requires_origin (See ??);
- step_edge_feature_shape_constraint (See ??);
- stratum_3d_shape_constraint (See ??);
- stratum_planar_shape_constraint (See ??);
- subtype_mandatory_shape_representation (See ??).

5.2.3.2.84 shape_representation_relationship

The base definition of the shape_representation_relationship entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of shape_representation_relationship is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the shape_representation_relationship entity:

- component_part_2d_geometric_representation_relationship_constraint (See 5.2.3.42) ;
- component_part_3d_geometric_representation_relationship_constraint (See 5.2.3.43) ;
- consistent_part_template_shape_projection_constraint (See ??) ;
- consistent_physical_unit_shape_projection_constraint (See ??);
- consistent_stratum_shape_projection_constraint (See ??).

5.2.3.2.85 specified_higher_usage_occurrence

The base definition of the specified_higher_usage_occurrence entity is given in ISO 10303-44. The following modifications apply to this Part of ISO 10303.

The definition of specified_higher_usage_occurrence is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the specified_higher_usage_occurrence entity:

- next_higher_assembly_interface_requirement_constraint (See ??) .

5.2.3.2.86 text_literal

The base definition of the text_literal entity is given in ISO 10303-46. The following modifications apply to this Part of ISO 10303.

The definition of text_literal is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the text_literal entity:

- subtype_mandatory_text_literal (See ??) .

5.2.3.2.87 tolerance_value

The base definition of the tolerance_value entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of tolerance_value is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the tolerance_value entity:

- dependent_instantiable_tolerance_value (See ??) .

5.2.3.2.88 tolerance_zone

The base definition of the tolerance_zone entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of tolerance_zone is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the tolerance_zone entity:

- tolerance_zone_constraint (See ??) .

5.2.3.2.89 tolerance_zone_definition

The base definition of the tolerance_zone_definition entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of tolerance_zone_definition is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the tolerance_zone_definition entity:

- conical_tolerance_zone_definition_constraint (See 5.2.3.67) ;
- tolerance_zone_definition_constraint (See ??) ;
- tolerance_zone_definition_with_per_unit_size_specification_constraint (See ??) .

5.2.3.2.90 tolerance_zone_form

The base definition of the tolerance_zone_form entity is given in ISO 10303-47. The following modifications apply to this Part of ISO 10303.

The definition of tolerance_zone_form is modified as follows:

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the tolerance_zone_form entity:

- tolerance_zone_form_constraint (See ??) .

5.2.3.2.91 versioned_action_request

The base definition of the versioned_action_request entity is given in ISO 10303-41. The following modifications apply to this Part of ISO 10303.

The definition of versioned_action_request is modified as follows:

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the versioned_action_request entity:

- change_request_unique_constraint (See 5.2.3.32) ;
- versioned_action_request_requires_approval (See ??) ;
- versioned_action_request_requires_date_or_date_and_time (See ??) ;
- versioned_action_request_requires_person_organization (See ??);
- versioned_action_request_requires_status (See ??);
- work_request_constraint (See ??);
- work_request_unique_constraint (See ??).

5.2.3.3 add_design_object_management_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The add_design_object_management_relationship_unique_constraint rule constrains the add_design_object_assignment and add_design_object_request_assignment population members, that fill the role of ARM add_design_object_management_relationship, so that the combination of ARM current_design and ARM current_design_object attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE add_design_object_management_relationship_unique_constraint FOR
  ( add_design_object_assignment, add_design_object_request_assignment );
LOCAL
  pdr_bag : BAG OF product_definition_relationship := [];
  pd_bag : BAG OF product_definition := [];
  adoa_bag : BAG OF add_design_object_assignment;
  adora_bag : BAG OF add_design_object_request_assignment;
  pass : BOOLEAN := TRUE;
  mdo_bag : BAG OF managed_design_object;
END_LOCAL;

REPEAT i := 1 to SIZEOF(add_design_object_assignment) by 1;
  REPEAT j := 1 TO SIZEOF(add_design_object_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP' IN
        TYPEOF(add_design_object_assignment[i].items[j]))
      AND (add_design_object_assignment[i].items[j].name =
        'design object addition') ) THEN
      IF EXISTS( add_design_object_assignment[i].items[j].
        related_product_definition )
        THEN
          IF( NOT( add_design_object_assignment[i].items[j].
            related_product_definition
              IN pd_bag ) ) THEN
            pd_bag := pd_bag +
              add_design_object_assignment[i].items[j].
                related_product_definition;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

REPEAT i := 1 to SIZEOF(add_design_object_request_assignment) by 1;
  REPEAT j := 1 TO
    SIZEOF(add_design_object_request_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP' IN
        TYPEOF(add_design_object_request_assignment[i].items[j]))
      AND (add_design_object_request_assignment[i].items[j].name =
        'design object addition') ) THEN
      IF EXISTS(
        add_design_object_request_assignment[i].items[j].
          related_product_definition )
        THEN
          IF( NOT(
            add_design_object_request_assignment[i].items[j].
              related_product_definition

```

```

                                IN pd_bag ) ) THEN
    pd_bag := pd_bag +
        add_design_object_request_assignment[i].items[j].
            related_product_definition;
    END_IF;
    END_IF;
    END_IF;
    END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pd_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    mdo_bag := [];
    pdr_bag := QUERY( pdr <* USEDIN(pd_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
        (pdr\product_definition_relationship.name = 'design object addition') );
    REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
        IF ( NOT pass ) THEN ESCAPE;
        END_IF;
        adoa_bag := QUERY( adoa <* add_design_object_assignment |
            (pdr_bag[j] IN adoa.items) );
        REPEAT k := 1 to SIZEOF(adoa_bag) by 1;
            IF ( NOT pass ) THEN ESCAPE;
            END_IF;
            REPEAT l := 1 to SIZEOF(adoa_bag[k].items) by 1;
                IF EXISTS( adoa_bag[k].items[l] ) THEN
                    IF ( adoa_bag[k].items[l] IN mdo_bag ) THEN
                        pass := FALSE;
                        ESCAPE;
                    ELSE
                        mdo_bag := mdo_bag + adoa_bag[k].items[l];
                    END_IF;
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
    REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
        IF ( NOT pass ) THEN ESCAPE;
        END_IF;
        adora_bag := QUERY( adora <* add_design_object_request_assignment |
            (pdr_bag[j] IN adora.items) );
        REPEAT k := 1 to SIZEOF(adora_bag) by 1;
            IF ( NOT pass ) THEN ESCAPE;
            END_IF;
            REPEAT l := 1 to SIZEOF(adora_bag[k].items) by 1;
                IF ( adora_bag[k].items[l] IN mdo_bag ) THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    mdo_bag := mdo_bag + adora_bag[k].items[l];
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

        END_IF;
    END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE

    WR1: pass;
END_RULE; -- add_design_object_management_relationship_unique_constraint
( *
```

Formal propositions:

WR1: Every instance of `add_design_object_assignment` and `add_design_object_request_assignment` that has a common `product_definition` that is the related `product_definition` of a `product_definition_relationship` that has a name of 'design object addition' where the `product_definition_relationship` is a member of the `items` attribute of the `add_design_object_assignment` and `add_design_object_request_assignment` must have unique managed `design object` entities in their `items` attributes.

5.2.3.4 adjacent_stratum_surface_definition_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `adjacent_stratum_surface_definition_constraint` rule constrains the related and relating `shape_aspect` of a `shape_aspect_relationship` when it is used as an adjacent stratum surface definition.

EXPRESS specification:

```

*)
RULE adjacent_stratum_surface_definition_constraint FOR
( shape_aspect_relationship );
WHERE

    WR1: SIZEOF (QUERY (sar <* shape_aspect_relationship |
                        (sar\shape_aspect_relationship.name = 'adjacent stratum surface definition'
                        (NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                            'STRATUM_SURFACE' IN TYPEOF (sar.related_shape_aspect)) AND
                            (sar.related_shape_aspect\shape_aspect.description IN ['secondary surface'])))))) = 0;

    WR2: SIZEOF (QUERY (sar <* shape_aspect_relationship |
                        (sar\shape_aspect_relationship.name = 'adjacent stratum surface definition'
                        (NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                            'STRATUM_SURFACE' IN TYPEOF (sar.relying_shape_aspect)) AND
                            (sar.relying_shape_aspect\shape_aspect.description IN ['primary surface'])))))) = 0;
END_RULE; -- adjacent_stratum_surface_definition_constraint
( *
```

Formal propositions:

WR1: If shape_aspect_relationship.name = 'adjacent stratum surface definition' the shape_aspect_relationship.related_shape_aspect shall be a stratum_surface with a description of 'secondary surface'.

WR2: If shape_aspect_relationship.name = 'adjacent stratum surface definition' the shape_aspect_relationship.relate_shape_aspect shall be a stratum_surface with a description of 'primary surface'.

5.2.3.5 adjacent_stratum_surface_definition_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The adjacent_stratum_surface_definition_unique_constraint rule constrains shape_aspect_relationship population members, that fill the role of ARM adjacent_stratum_surface_definition, so that the ARM name shall be unique and that the combination of ARM precedent_surface and ARM subsequent_surface attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE adjacent_stratum_surface_definition_unique_constraint FOR
  ( shape_aspect_relationship );
LOCAL
  assd : BAG OF shape_aspect_relationship :=
    QUERY( sar <* shape_aspect_relationship
  | (sar\shape_aspect_relationship.description = 'adjacent stratum surface definition') );
  pass1 : BOOLEAN := TRUE;
  name_bag : BAG OF STRING := [];
  pss_bag : BAG OF stratum_surface := [];
  sar_bag : BAG OF shape_aspect_relationship;
  pass2 : BOOLEAN := TRUE;
  sss_bag : BAG OF stratum_surface;
END_LOCAL;

REPEAT i := 1 to SIZEOF(assd) by 1;
  IF EXISTS( assd[i]\shape_aspect_relationship.name ) THEN
    IF ( assd[i]\shape_aspect_relationship.name IN name_bag ) THEN
      pass1 := FALSE;
      ESCAPE;
    ELSE
      name_bag := name_bag + assd[i]\shape_aspect_relationship.name;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(assd) by 1;
  IF EXISTS( assd[i].relating_shape_aspect ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_SURFACE' IN TYPEOF(assd[i].relating_shape_aspect) ) THEN
      IF( NOT( assd[i].relating_shape_aspect IN pss_bag ) ) THEN
        pss_bag := pss_bag + assd[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pss_bag) by 1;
  IF ( NOT pass2 ) THEN ESCAPE;
  END_IF;
  sss_bag := [];
  sar_bag := QUERY( sar <* assd | (sar.relating_shape_aspect ==:
    pss_bag[i]) );
  REPEAT j := 1 to SIZEOF(sar_bag) by 1;
    IF EXISTS( sar_bag[j].related_shape_aspect ) THEN
      IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_SURFACE' IN TYPEOF(assd[j].related_shape_aspect) ) THEN
        IF ( sar_bag[j].related_shape_aspect IN sss_bag ) THEN
          pass2 := FALSE;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

```

```

        ESCAPE;
    ELSE
        sss_bag := sss_bag + sar_bag[j].related_shape_aspect;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
WHERE

    WR1: pass1;

    WR2: pass2;
END_RULE; -- adjacent_stratum_surface_definition_unique_constraint
(*

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a description of 'adjacent stratum surface definition' must have a unique name attribute.

WR2: Every instance of shape_aspect_relationship with a description of 'adjacent stratum surface definition' must have a unique combination of relating_shape_aspect and related_shape_aspect attributes.

5.2.3.6 aggregate_connectivity_requirement_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The aggregate_connectivity_requirement_unique_constraint rule constrains product_definition_relationship population members, that fill the role of ARM aggregate_connectivity_requirement, so that the ARM design_definition_path attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE aggregate_connectivity_requirement_unique_constraint FOR
  ( product_definition_relationship );
LOCAL
  acr : BAG OF product_definition_relationship :=
    QUERY( pdr <* product_definition_relationship |
    (pdr\product_definition_relationship.name = 'aggregate connectivity require-
  pass : BOOLEAN := TRUE;
  pd_bag : BAG OF product_definition := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(acr) by 1;
  IF EXISTS( acr[i].related_product_definition ) THEN
    IF ( acr[i].id = 'design composition path' ) THEN
      IF ( acr[i].related_product_definition IN pd_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        pd_bag := pd_bag + acr[i].related_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- aggregate_connectivity_requirement_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of `product_definition_relationship` with a name of 'aggregate connectivity requirement' must have a unique `related_product_definition` attribute that points to a `product_definition` with an id of 'design composition path'.

5.2.3.7 analytical_model_port_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `analytical_model_port_unique_constraint` rule constrains the `analytical_model_port` population members, that fill the role of ARM `analytical_model_port`, so that the combination of ARM `port_name` and ARM `accessed_analytical_model` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE analytical_model_port_unique_constraint FOR
  ( analytical_model_port );
LOCAL
  name_bag : BAG OF STRING := [];
  amp_bag : BAG OF analytical_model_port;
  rr_bag : BAG OF representation_relationship;
  pass : BOOLEAN := TRUE;
  am_bag : BAG OF analytical_model;
END_LOCAL;

REPEAT i := 1 to SIZEOF(analytical_model_port) by 1;
  IF EXISTS( analytical_model_port[i].name ) THEN
    IF( NOT( analytical_model_port[i].name IN name_bag ) ) THEN
      name_bag := name_bag + analytical_model_port[i].name;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(name_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  amp_bag := QUERY( amp <* analytical_model_port |
    (amp\representation.name = name_bag[i]) );
  am_bag := [];
  REPEAT j := 1 to SIZEOF(amp_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    rr_bag := QUERY( rr <* USEDIN( amp_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'REPRESENTATION_RELATIONSHIP.REP_2' ) | ((rr\representation_relationship.name = 'access r
AND
( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'ANALYTICAL_MODEL' IN TYPEOF(rr.rep_1))) );
    REPEAT k := 1 to SIZEOF(rr_bag) by 1;
      IF EXISTS( rr_bag[k].rep_1 ) THEN
        IF ( rr_bag[k].rep_1 IN am_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          am_bag := am_bag + rr_bag[k].rep_1;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;
WHERE

```

```

    WR1: pass;
END_RULE; -- analytical_model_port_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of `analytical_model_port` must have a unique combination of name attribute and `analytical_model` entity assigned to the `analytical_model_port` by a `representation_relationship` with a name of 'access mechanism'.

5.2.3.8 `analytical_model_vector_port_assignment_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `analytical_model_vector_port_assignment_constraint` rule constrains the definition of the `property_definition_representation` to ensure that there will be an ordered list of `assigned_functional_unit_terminals`.

EXPRESS specification:

```

*)
( *
RULE analytical_model_vector_port_assignment_constraint FOR
(property_definition_representation);
END_RULE; -- analytical_model_vector_port_assignment_constraint
*)
( *

```

Informal propositions:

IP1: IF the `property_definition_representation.used_representation` [is an `analytical_model_port`] [is an items of a `group_assignment` that has a name = ('vector port') ('digital vector port')] THEN `property_definition_representation.definition.description` shall be an integer greater than or equal to 0.

IP2: IF the `property_definition_representation.used_representation` [is an `analytical_model_port`] [is an items of a `group_assignment` that has a name = ('vector port') ('digital vector port')] THEN the combination of the `property_definition_representation.used_representation` and `property_definition_representation.definition.description` shall be unique.

5.2.3.9 `angular_dimension_with_direction_vector_unique_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `angular_dimension_with_direction_vector_unique_constraint` rule constrains the `angular_dimension_with_orientation` population members, that fill the role of ARM `angular_dimension_with_direction_vector`, so that the ARM `measurement_orientation` attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE angular_dimension_with_direction_vector_unique_constraint FOR
  ( angular_dimension_with_orientation );
LOCAL
  p_bag : BAG OF property_definition;
  pdr_bag : BAG OF property_definition_relationship;
  pass : BOOLEAN := TRUE;
  pd_bag : BAG OF property_definition := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(angular_dimension_with_orientation) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  p_bag := QUERY( pd <* USEDIN(angular_dimension_with_orientation[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (pd\property_definition.description =
'dimensional location property') );
  REPEAT j := 1 to SIZEOF(p_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    pdr_bag := QUERY( pdr <* USEDIN(p_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
((pdr\property_definition_relationship.name = 'measurement orientation') AND
(pdr.related_property_definition\property_definition.description =
'datum based vector orientation')) );
    REPEAT k := 1 to SIZEOF(pdr_bag) by 1;
      IF EXISTS( pdr_bag[k].related_property_definition ) THEN
        IF ( pdr_bag[k].related_property_definition IN pd_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pd_bag := pd_bag + pdr_bag[k].related_property_definition;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- angular_dimension_with_direction_vector__unique_constraint
(

```

Formal propositions:

WR1: Every instance of `angular_dimension_with_orientation` must have a unique `property_definition` entity with a name of 'datum based vector orientation' that is related to the `angular_dimension_with_orientation` by a `property_definition_relationship` with a name of 'measurement orientation'.

5.2.3.10 `angular_size_dimension_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `angular_size_dimension_constraint` rule constrains the use of an `angular_size` when it is used as an angular size dimension.

EXPRESS specification:

```
* )
( *
RULE angular_size_dimension_constraint FOR
    (angular_size);
END_RULE; -- angular_size_dimension_constraint
* )
( *
```

Informal propositions:

IP1: Each `angular_size` shall be represented by exactly zero `shape_dimension_representation` that contain at least one `representation_item` that has a name that is neither 'full angle' nor 'half angle'.

IP2: Each `angular_size` shall be represented by exactly zero `shape_dimension_representation` that contains a `representation_item` with the name of 'full angle' and a `representation_item` with a name of 'half angle'.

IP3: Each `angular_size` shall have a name of 'angular'.

5.2.3.11 `application_context_requires_ap_definition`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `application_context_requires_ap_definition` rule assures that each instance of `application_context` has an `application_protocol_definition` with name equal to 'electronic_assembly_interconnect_and_packaging_design'.

EXPRESS specification:

```

*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE

  WR1: SIZEOF (QUERY (ac <* application_context |
    NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
      (ac :=: apd\application_protocol_definition.application)
      AND
      (apd\application_protocol_definition.application_interpreted_model_schema_name =
        'electronic_assembly_interconnect_and_packaging_design'
      ))) = 1 ))) = 0;
END_RULE; --application_context_requires_ap_definition
( *

```

Formal propositions:

WR1: For each instance of application_context, there shall be exactly one instance of application_protocol_definition that references the instance of application_context as its application with a value of 'electronic_assembly_interconnect_and_packaging_design' as its application_interpreted_model_schema_name.

5.2.3.12 approval_requires_approval_date_time

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The approval_requires_approval_date_time rule assures that each instance of approval has an associated date or time.

EXPRESS specification:

```

*)
RULE approval_requires_approval_date_time FOR (approval,
  approval_date_time);
WHERE

  WR1: SIZEOF(QUERY ( app <* approval |
    NOT (SIZEOF (QUERY (adt <* approval_date_time |
      app :=: adt.dated_approval))) = 1))) = 0;
END_RULE; -- approval_requires_approval_date_time
( *

```

Formal propositions:

WR1: For each instance of approval, there shall be exactly one instance of approval_date_time.

5.2.3.13 approval_requires_approval_person_organization

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The approval_requires_approval_person_organization rule assures that for each instance of approval there is an organization or person that is responsible for that approval.

EXPRESS specification:

```
*)
RULE approval_requires_approval_person_organization FOR (approval,
    approval_person_organization);
WHERE

    WR1: SIZEOF (QUERY (app <* approval |
        NOT (SIZEOF (QUERY (apo <* approval_person_organization |
            app :=: apo.authorized_approval)) >= 1))) = 0;
END_RULE; -- approval_requires_approval_person_organization
(*
```

Formal propositions:

WR1: For each instance of approval, there shall be one or more instances of approval_person_organization.

5.2.3.14 approval_role_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The approval_role_constraint rule assures that each approval_role has a role that is 'authorizer'.

EXPRESS specification:

```
*)
RULE approval_role_constraint FOR
    (approval_role);
WHERE
```

```

WR1: SIZEOF (QUERY (apr <* approval_role |
    NOT (apr\approval_role.role = 'authorizer')
    )) = 0;
END_RULE; -- approval_role_constraint
(*)

```

Formal propositions:

WR1: For each instance of approval_role, the role shall be 'authorizer'.

5.2.3.15 approvals_are_assigned

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The approvals_are_assigned rule assures that each approval is referenced by at least one approval_assignment as assigned_approval.

EXPRESS specification:

```

*)
RULE approvals_are_assigned FOR
    (approval, approval_assignment);
WHERE

    WR1: SIZEOF (QUERY (app <* approval |
        NOT (SIZEOF (QUERY (aa <* approval_assignment |
            app := aa.assigned_approval )) >= 1 ))) = 0;
END_RULE; -- approvals_are_assigned
(*)

```

Formal propositions:

WR1: For each instance of approval, there shall be one or more instances of approval_assignment.

5.2.3.16 assembly_alternate_product_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The assembly_alternate_product_constraint rule constrains the base of alternate_product_relationship when it is used as an assembly alternate product.

EXPRESS specification:

```
* )
( *
RULE assembly_alternate_product_constraint FOR
(alternate_product_relationship);
END_RULE; --assembly_alternate_product_constraint
* )
( *
```

Informal propositions:

IP1: IF alternate_product_relationship.basis = 'assembly alternate product' THEN alternate_product_relationship.base shall be a product which is the related_product_definition of an assembly_component_usage whose relating_product_definition {is a physical_unit} {has a frame_of_reference of an application_context_element where application_context_element.name = 'physical design'}.

5.2.3.17 assembly_composition_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The assembly_composition_relationship_unique_constraint rule constrains assembly_component_usage population members, that fill the role of ARM assembly_composition_relationship, so that the combination of ARM assembly and ARM component attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE assembly_composition_relationship_unique_constraint FOR
  ( assembly_component_usage );
LOCAL
  acr : BAG OF assembly_component_usage :=
  QUERY( acu <* assembly_component_usage |
(acu\product_definition_relationship.name = 'assembly composition') );
  pu_bag : BAG OF physical_unit := [];
  acu_bag : BAG OF assembly_component_usage;
  pass : BOOLEAN := TRUE;
  cd_bag : BAG OF component_definition;
END_LOCAL;

REPEAT i := 1 to SIZEOF(acr) by 1;
  IF EXISTS( acr[i].relating_product_definition ) THEN
    IF( (acr[i].relating_product_definition.frame_of_reference.name =
      'physical design')
  AND ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT' IN TYPEOF(acr[i].relating_product_definition)) ) THEN
      IF ( NOT ( acr[i].relating_product_definition IN pu_bag ) ) THEN
        pu_bag := pu_bag + acr[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pu_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  cd_bag := [];
  acu_bag := QUERY( acu <* acr | (acu.relating_product_definition :=:
    pu_bag[i]) );
  REPEAT j := 1 to SIZEOF(acu_bag) by 1;
    IF EXISTS( acu_bag[j].related_product_definition ) THEN
      IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION' IN
        TYPEOF(acu_bag[j].related_product_definition) ) THEN
        IF ( acu_bag[j].related_product_definition IN cd_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          cd_bag := cd_bag + acu_bag[j].related_product_definition;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

```

```

    WR1: pass;
END_RULE; -- assembly_composition_relationship_unique_constraint
( *
```

Formal propositions:

WR1: Every instance of `assembly_component_usage` with a name of 'assembly composition' must have a unique combination of `related_product_definition` and `relating_product_definition` attributes.

5.2.3.18 assembly_module_usage_view_connector_relationship_unique_constraint

The `assembly_module_usage_view_connector_relationship_unique_constraint` rule constrains `assembly_module_usage_view_connector_relationship` population members, that fill the role of ARM `assembly_module_usage_view_connector_relationship`, so that the combination of ARM `externally_visible_partial_reference_designation` and ARM `associating_usage` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE assembly_module_usage_view_connector_relationship_unique_constraint FOR
  ( assembly_module_usage_view_connector_relationship );
LOCAL
  r_bag : BAG OF representation;
  pdr_bag : BAG OF property_definition_representation;
  pu_bag : BAG OF physical_unit := [];
  pd_bag : BAG OF property_definition;
  amuvcr : BAG OF assembly_module_usage_view_connector_relationship;
  rr_bag : BAG OF representation_relationship;
  pass : BOOLEAN := TRUE;
  desc_bag : BAG OF STRING;
END_LOCAL;

REPEAT i := 1 to
  SIZEOF(assembly_module_usage_view_connector_relationship) by 1;
  r_bag := USEDIN(assembly_module_usage_view_connector_relationship[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REPRESENTATION.ITEMS');

  REPEAT j := 1 to SIZEOF(r_bag) by 1;
    pdr_bag := QUERY( pdr <* USEDIN(r_bag[j],
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | (
  ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PHYSICAL_UNIT' IN TYPEOF(pdr.definition.definition)) AND
  (pdr.definition.definition.frame_of_reference.name =
    'physical design usage') AND
  (pdr.definition.definition\product_definition.name = 'assembly module')
  ));
    REPEAT k := 1 to SIZEOF(pdr_bag) by 1;
      IF EXISTS( pdr_bag[k].definition.definition ) THEN
        IF( NOT( pdr_bag[k].definition.definition IN pu_bag ) ) THEN
          pu_bag := pu_bag + pdr_bag[k].definition.definition;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pu_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  desc_bag := [];
  pd_bag := USEDIN(pu_bag[i],
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PROPERTY_DEFINITION.DEFINITION');
  REPEAT j := 1 to SIZEOF(pd_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
  END_REPEAT;
END_REPEAT;

```

```

    pdr_bag := USEDIN( pd_bag[j],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
    REPEAT k := 1 to SIZEOF(pdr_bag) by 1;
        IF ( NOT pass ) THEN ESCAPE;
        END_IF;
        amuvcr :=
            QUERY( r <* assembly_module_usage_view_connector_relationship |
(r IN pdr_bag[k].used_representation.items) );
        REPEAT l := 1 to SIZEOF(amuvcr) by 1;
            IF ( NOT pass ) THEN ESCAPE;
            END_IF;
            r_bag := USEDIN( amuvcr[l],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION.ITEMS' );
            REPEAT m := 1 to SIZEOF(r_bag) by 1;
                IF ( NOT pass ) THEN ESCAPE;
                END_IF;
                rr_bag := QUERY( rr <* USEDIN(r_bag[m],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REPRESENTATION_RELATIONSHIP.REP_2') |
((rr\representation_relationship.name = 'reference designation assignment') AND
(rr.rep_1\representation.name = 'externally visible partial reference designation') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(rr.rep_1.items[1])) AND
(rr.rep_1.items[1]\representation_item.name = 'partial reference designation')) );
                REPEAT n := 1 to SIZEOF(rr_bag) by 1;
                    IF EXISTS( rr_bag[n].rep_1.items[1]\descriptive_representation_item.description
                    IF ( rr_bag[n].rep_1.items[1]\descriptive_representation_item.description IN
                        pass := FALSE;
                        ESCAPE;
                    ELSE
                        desc_bag := desc_bag + rr_bag[n].rep_1.items[1]\descriptive_representation.
                    END_IF;
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    WHERE

    WR1: pass;
END_RULE; -- assembly_module_usage_view_connector_relationship_-
-- unique_constraint
( *
```

Formal propositions:

WR1: Every instance of `assembly_module_usage_view_connector_relationship` must have a unique combination of `physical_unit` entity and `descriptive_representation_item` description attribute where the `physical_unit` is related to the `assembly_module_usage_view_connector_relationship` by a `property_definition_representation` and the `descriptive_representation_item` has a name of 'partial reference designation' and is related to the `assembly_module_usage_view_connector_relationship` by a `representation_relationship` with a name of 'reference designation assignment'.

5.2.3.19 assembly_requirement_allocation_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `assembly_requirement_allocation_constraint` rule constrains the related and relating `shape_aspect` or `product_definition` when the `shape_aspect_relationship` or the `product_definition_relationship` is being used as an assembly requirement.

EXPRESS specification:

```
* )
( *
RULE assembly_requirement_allocation_constraint FOR
( shape_aspect_relationship,
  product_definition_relationship );
END_RULE; -- assembly_requirement_allocation_constraint
* )
( *
```

Informal propositions:

IP1: IF `shape_aspect_relationship.name` = 'assembly requirement allocation' the `related_shape_aspect` must be a (`component_terminal`) (`component_interface_terminal`) (`component_shape_aspect` whose description is ('component feature') ('component termination passage interface terminal') ('land interface terminal') ('non functional land interface terminal') ('printed connector component interface terminal')) or it shall be an `assembly_joint`.

IP2: IF `shape_aspect_relationship.name` = 'assembly requirement allocation' the `relating_shape_aspect` shall have a {`of_shape` with a `property_definition` with a definition that is a `product_definition` with a `frame_of_reference` that is a `application_context_element` with a name of 'assembly requirement'} and a definition for a `property_definition` that is a `requirements_property`.

IP3: IF `product_definition_relationship.name` = 'assembly requirement allocation' the `related_product_definition` shall be a `component_definition`.

IP4: IF `product_definition_relationship.name = 'assembly requirement allocation'` the `relating_product_definition` shall have a `{frame_of_reference}` with a `that` is a `application_context_element` with a name of `'assembly requirement'` and a definition for a `property_definition` that is a `requirements_property`.

5.2.3.20 `assembly_separation_requirement_allocation_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `assembly_separation_requirement_allocation_constraint` rule constrains the `related_shape_aspect` of a `shape_aspect_relationship` when it is being used as an assembly separation requirement.

EXPRESS specification:

```
*)
(*
RULE assembly_separation_requirement_allocation_constraint FOR
( shape_aspect_relationship );
END_RULE; -- assembly_separation_requirement_allocation_constraint
*)
(*
```

Informal propositions:

IP1: IF `shape_aspect_relationship.name = 'assembly separation requirement'` the `related_shape_aspect` shall be a `(component_terminal)` `(component_interface_terminal)` `(component_shape_aspect` whose description is `('component feature')` `('component termination passage interface terminal')` `('land interface terminal')` `('non functional land interface terminal')` `('printed connector component interface terminal')`) or the `related_shape_aspect` shall be an `assembly_joint`.

5.2.3.21 `bare_die_unique_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `bare_die_unique_constraint` rule constrains the `bare_die`, `externally_defined_bare_die` and `library_defined_bare_die` population members, that fill the role of ARM `bare_die`, so the combination of ARM `implemented_function` and ARM `used_package` shall be unique within the population.

EXPRESS specification:

```

*)
RULE bare_die_unique_constraint FOR
  ( bare_die, externally_defined_bare_die, library_defined_bare_die );
LOCAL
  bd : BAG OF physical_unit := QUERY( r <* bare_die |
(r.frame_of_reference.name = 'physical design usage' ) );
  edbd : BAG OF physical_unit := QUERY( r <* externally_defined_bare_die |
(r.frame_of_reference.name = 'physical design usage' ) );
  ldbd : BAG OF physical_unit := QUERY( r <* library_defined_bare_die |
(r.frame_of_reference.name = 'physical design usage' ) );
  pu : BAG OF physical_unit;
  pdr_bag : BAG OF product_definition_relationship;
  fu_bag : BAG OF functional_unit := [];
  pu_bag : BAG OF physical_unit;
  ut_bag : BAG OF property_definition;
  pass : BOOLEAN := TRUE;
  pd_bag : BAG OF property_definition;
END_LOCAL;

pu := bd + edbd + ldbd;
REPEAT i := 1 to SIZEOF(pu) by 1;
  pdr_bag := QUERY( pdr <* USEDIN( pu[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION' ) |
( (pdr\product_definition_relationship.name = 'implemented function') AND
(pdr.relating_product_definition.frame_of_reference.name =
'functional design usage')
AND ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT' IN TYPEOF(pdr.relating_product_definition)) ) );
  REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
    IF EXISTS( pdr_bag[j].relating_product_definition ) THEN
      IF( NOT( pdr_bag[j].relating_product_definition IN fu_bag ) ) THEN
        fu_bag := fu_bag + pdr_bag[j].relating_product_definition;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(fu_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  pu_bag := QUERY( r <* pu | (SIZEOF(QUERY(pdr <* USEDIN( fu_bag[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION' ) |
((pdr\product_definition_relationship.name = 'implemented function')AND
(pdr.related_product_definition :=: r)) ))>0 ) );
  pd_bag := [];
  REPEAT j := 1 to SIZEOF(pu_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;

```

```

    END_IF;
    ut_bag := QUERY( pd <* USEDIN( pu_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION' ) | (pd\property_definition.name = 'unit technology') );
    REPEAT k := 1 to SIZEOF(ut_bag) by 1;
        IF EXISTS( ut_bag[k] ) THEN
            IF ( ut_bag[k] IN pd_bag ) THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                pd_bag := pd_bag + ut_bag[k];
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE

    WR1: pass;
END_RULE; -- bare_die_unique_constraint
( *
```

Formal propositions:

WR1: Every instance of bare_die, externally_defined_bare_die and library_defined_bare_die must have a unique combination of functional_unit entity related by an 'implemented function' product_definition_relationship and a property_definition entity with a name of 'unit technology' whose definition attribute points to the bare_die, externally_defined_bare_die and library_defined_bare_die.

5.2.3.22 bevel_edge_feature_shape_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The bevel_edge_feature_shape_constraint rule requires maximum and minimum bevel width and bevel angles to be assigned to a shape_representation when it is being used as a bevel edge feature shape.

EXPRESS specification:

```

*)
( *
RULE bevel_edge_feature_shape_constraint FOR ( shape_representation );
END_RULE; -- bevel_edge_feature_shape_constraint
*)
( *
```


Informal propositions:

IP1: For each shape_representation representation.name = 'bevel edge feature shape' there shall be exactly one representation_item that is a length_measure_with_unit where representation_item.name = 'maximum bevel width' assigned to representation.items.

IP2: For each shape_representation representation.name = 'bevel edge feature shape' there shall be exactly one representation_item which is a length_measure_with_unit where representation_item.name = 'minimum bevel width' assigned to representation.items.

IP3: For each shape_representation representation.name = 'bevel edge feature shape' there shall be exactly one representation_item which is a plane_angle_measure_with_unit where representation_item.name = 'maximum bevel angle' assigned to representation.items.

IP4: For each shape_representation representation.name = 'bevel edge feature shape' there shall be exactly one representation_item which is a plane_angle_measure_with_unit where representation_item.name = 'minimum bevel angle' assigned to representation.items.

IP5: This representation shall only be a representation of the shape of an edge segment of the interconnect product.

5.2.3.23 boundary_size_characteristic_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The boundary_size_characteristic_constraint rule constrains the use of a representation when it is used as a boundary size characteristic.

EXPRESS specification:

```

*)
RULE boundary_size_characteristic_constraint FOR
  (representation);
WHERE

  WR1: SIZEOF(QUERY(rep <* representation |
    ((rep\representation.name = 'diametrical boundary size')
    OR (rep\representation.name = 'opposing boundary set size'))
    AND (NOT(SIZEOF(QUERY(lmwu <* rep.items |
      (lmwu\representation_item.name = 'tolerance value')) = 1)))) = 0;
END_RULE; -- boundary_size_characteristic_constraint
(

```

Formal propositions:

WR1: Each representation with a name of either 'diametrical boundary size' or 'opposing boundary set size' shall contain exactly one length_measure_with_unit that has a name of 'tolerance value'.

Informal propositions:

IP1: Each representation with a name of either 'diametrical boundary size' or 'opposing boundary set size' shall contain exactly one `length_measure_with_unit` that has a name of 'tolerance value', and that instance of `length_measure_with_unit` shall be the same instance of `length_measure_with_unit` as specified as the magnitude by the `physical_unit_geometric_tolerance` that is specified as the defining tolerance by the `tolerance_zone` that is specified as the zone by the `tolerance_zone_definition` that specifies the `tolerance_zone_boundary` as its boundaries and the `property_definition` of which has a description of 'boundary zone definition with specified size' and is represented by the instance of representation with a name of either 'diametrical boundary size' or 'opposing boundary set size'.

5.2.3.24 boundary_zone_definition_with_specified_size_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `boundary_zone_definition_with_specified_size_constraint` rule constrains the use of a `property_definition` when it is used as a boundary zone definition with specified size.

EXPRESS specification:

```

*)
( *
RULE boundary_zone_definition_with_specified_size_constraint FOR
  (property_definition);
END_RULE; -- boundary_zone_definition_with_specified_size_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'boundary zone definition with specified size' shall be a `property_definition` of a `tolerance_zone_boundary`.

IP2: Each instance of `property_definition` that has a description of 'boundary zone definition with specified size' shall have exactly one representation with a name of either 'diametrical boundary size' or 'opposing boundary set size'.

5.2.3.25 bus_structural_definition_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `bus_structural_definition_unique_constraint` rule constrains the `product.name` population members to satisfy the uniqueness requirement of the ARM application object `Bus_structural_definition`.

EXPRESS specification:

```

*)

RULE bus_structural_definition_unique_constraint FOR
(bus_structural_definition, product_definition_formation, product);

LOCAL
    pass : BOOLEAN := TRUE;
    name_bag : BAG OF STRING := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(bus_structural_definition) by 1;
    IF EXISTS( bus_structural_definition[i].formation.of_product.name ) THEN
        IF (bus_structural_definition[i].formation.of_product.name IN
            name_bag) THEN

            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag +
bus_structural_definition[i]\product_definition.formation.of_product.name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE

WR1: pass;

END_RULE; -- bus_structural_definition_unique_constraint

( *
```

Formal propositions:

WR1: The product.name shall be unique within a population of bus_structural_definition.

5.2.3.26 cartesian_coordinate_system_constraint

The cartesian_coordinate_system_constraint rule requires the units of a global_unit_assigned_context to at least include a length_unit and a plane_angle_unit when the global_unit_assigned_context is also a geometric_representation_context and is a global_uncertainty_assigned_context.

EXPRESS specification:

```

*)
RULE cartesian_coordinate_system_constraint FOR
    (global_unit_assigned_context,
     geometric_representation_context,
     global_uncertainty_assigned_context);
LOCAL
    guac_inst : SET OF global_unit_assigned_context :=
        QUERY ( guac <* global_unit_assigned_context |
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF (guac)) AND
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT' IN TYPEOF (guac)));
END_LOCAL;
WHERE

WR1: SIZEOF (QUERY (guac <* guac_inst |
    NOT (SIZEOF (guac.units) <= 3))) = 0;

WR2: SIZEOF (QUERY (guac <* guac_inst | (NOT (
    (SIZEOF (QUERY (u <* guac.units |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.LENGTH_UNIT'
    IN TYPEOF (u)))) = 1)
    AND
    (SIZEOF (QUERY (u <* guac.units |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLANE_ANGLE_UNIT'
    IN TYPEOF (u)))) = 1)
    AND
    (SIZEOF (QUERY (u <* guac.units |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SOLID_ANGLE_UNIT'
    IN TYPEOF (u)) )) <= 1)
    )))) = 0;

WR3: SIZEOF( QUERY( grc <* guac_inst |
    NOT( (grc.coordinate_space_dimension = 2) OR
    (grc.coordinate_space_dimension = 3)))) = 0;

WR4: SIZEOF (QUERY (guac <* guac_inst |
    NOT ((SIZEOF (QUERY (u <* guac.uncertainty |
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF (u))) = 1)))) = 0;

```

```

WR5: sizeof (QUERY (guac <* guac_inst |
    NOT (sizeof (QUERY (u <* guac.units |NOT(
    NOT((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLANE_ANGLE_UNIT' IN TYPEOF(u)) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONVERSION_BASED_UNIT' IN TYPEOF(u)) AND
    (u\conversion_based_unit.name = 'degree'))
OR
    ((abs(u\conversion_based_unit.
        conversion_factor\measure_with_unit.value_component -
        0.0174532925) <= 0.001) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLANE_ANGLE_MEASURE_WITH_UNIT'
    IN TYPEOF(u\conversion_based_unit.conversion_factor)))
    ))) = 0)
    )) = 0;
END_RULE; -- cartesian_coordinate_system_constraint
(*

```

Formal propositions:

WR1: For each instance of `global_unit_assigned_context` implementing the arm `Cartesian_coordinate_system`, the set of units shall contain a maximum of 3 elements.

WR2: For each instance of `global_unit_assigned_context` implementing the arm `Cartesian_coordinate_system`, exactly one element of the set of units shall be a `length_unit`, exactly one element of the set of units shall be a `plane_angle_unit` where the unit is either radian or degree, and no more than one element of the set of units shall be a `solid_angle_unit`.

WR3: Each `geometric_representation_context` implementing the arm `Cartesian_coordinate_system`, the `coordinate_space_dimension` shall have a value of 2 or 3.

WR4: For each `global_uncertainty_assigned_context` implementing the arm `Cartesian_coordinate_system`, exactly one element of the set of uncertainty shall be a `length_measure_with_unit`.

5.2.3.27 cartesian_transformation_operator_2d_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `cartesian_transformation_operator_2d_constraint` rule requires the units of a `global_unit_assigned_context` to at least include a `length_unit` and a `plane_angle_unit` and a scale greater than zero.

EXPRESS specification:

```
*)  
RULE cartesian_transformation_operator_2d_constraint FOR  
      (cartesian_transformation_operator_2d);  
WHERE  
  
      WR1: SIZEOF(QUERY ( cto2 <* cartesian_transformation_operator_2d | NOT(  
        EXISTS(cto2.axis1) AND EXISTS(cto2.axis2)  
        AND (cto2.scale > 0.0 )))) = 0;  
END_RULE; -- cartesian_transformation_operator_2d_constraint  
(*
```

Formal propositions:

WR1: Axis1 and axis2 shall exist and the scale value shall be greater than 0.0.

5.2.3.28 centre_of_symmetry_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The centre_of_symmetry_constraint rule constrains the use of a centre_of_symmetry when it is used as a centre of symmetry.

EXPRESS specification:

```
*)  
(*  
RULE centre_of_symmetry_constraint FOR  
      (centre_of_symmetry);  
END_RULE; -- centre_of_symmetry_constraint  
*)  
(*
```

Informal propositions:

IP1: Each centre_of_symmetry shall have description of either 'axis', 'plane', or 'point'.

5.2.3.29 certification_requires_date_or_date_and_time

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The certification_requires_date_or_date_and_time rule assures that each certification has an associated applied_date_assignment or an applied_date_and_time_assignment.

EXPRESS specification:

```

*)
RULE certification_requires_date_or_date_and_time FOR
  (certification, applied_date_and_time_assignment,
   applied_date_assignment);
WHERE

  WR1: SIZEOF (QUERY (cert <* certification |
    NOT ((SIZEOF (QUERY (adata <* applied_date_and_time_assignment |
      cert IN adata.items )) = 1 ) OR
      (SIZEOF (QUERY (ada <* applied_date_assignment |
        cert IN ada.items )) = 1 )))) = 0;
END_RULE; -- certification_requires_date_or_date_and_time
( *

```

Formal propositions:

WR1: For each instance of certification, exactly one instance of applied_date_and_time_assignment or exactly one instance of applied_date_assignment shall contain the instance of certification in its set of items.

5.2.3.30 certification_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The certification_unique_constraint rule constrains the certification.name population members to satisfy the uniqueness requirement of the ARM application object Certification.

EXPRESS specification:

```

*)

RULE certification_unique_constraint FOR (certification);

LOCAL
  r : BAG OF certification := QUERY (r <* certification | TRUE);
  pass : BOOLEAN := TRUE;
  name_bag : BAG OF string := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF (r) by 1;
  IF EXISTS( r[i]\certification.name ) THEN
    IF (r[i]\certification.name IN name_bag) THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      name_bag := name_bag + r[i]\certification.name;
    END_IF;
  END_IF;
END_REPEAT;

WHERE

  WR1: pass;

END_RULE; -- certification_unique_constraint

( *

```

Formal propositions:

WR1: The name shall be unique within a population of certification.

5.2.3.31 change_design_object_management_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The change_design_object_management_relationship_unique_constraint rule constrains the change_from_design_object_assignment, change_from_design_object_request_assignment, change_to_design_object_assignment and change_to_design_object_request_assignment population members, that fill the role of ARM change_design_object_management_relationship, so that the combination of ARM current_design and ARM current_design_object attributes shall be unique within the population and that the combination of ARM previous_design and ARM previous_design_object attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE change_design_object_management_relationship_unique_constraint FOR
( change_from_design_object_assignment,
  change_from_design_object_request_assignment,
  change_to_design_object_assignment,
  change_to_design_object_request_assignment );
LOCAL
  pd_bag : BAG OF product_definition := [];
  pdr_bag : BAG OF product_definition_relationship;
  ctdoa_bag : BAG OF change_to_design_object_assignment;
  ctдора_bag : BAG OF change_to_design_object_request_assignment;
  cfdoa_bag : BAG OF change_from_design_object_assignment;
  cfдора_bag : BAG OF change_from_design_object_request_assignment;
  pass1 : BOOLEAN := TRUE;
  pass2 : BOOLEAN := TRUE;
  mdo_bag : BAG OF managed_design_object;
END_LOCAL;

REPEAT i := 1 to SIZEOF(change_from_design_object_assignment) by 1;
  REPEAT j := 1 TO
    SIZEOF(change_from_design_object_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP' IN
      TYPEOF(change_from_design_object_assignment[i].items[j]))
      AND (change_from_design_object_assignment[i].items[j].name =
        'design object change') ) THEN
      IF EXISTS
        ( change_from_design_object_assignment[i].items[j].
          relating_product_definition )
        THEN
          IF( NOT
            ( change_from_design_object_assignment[i].items[j].
              relating_product_definition
              IN pd_bag ) ) THEN
            pd_bag := pd_bag +
              change_from_design_object_assignment[i].items[j].
                relating_product_definition;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

REPEAT i := 1 to
  SIZEOF(change_from_design_object_request_assignment) by 1;
  REPEAT j := 1 TO
    SIZEOF(change_from_design_object_request_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP' IN

```

```

        TYPEOF(change_from_design_object_request_assignment[i].items[j]))
AND (change_from_design_object_request_assignment[i].items[j].name =
        'design object change') ) THEN
    IF EXISTS
    ( change_from_design_object_request_assignment[i].items[j].
        relating_product_definition )
        THEN
            IF( NOT
            ( change_from_design_object_request_assignment[i].items[j].
                relating_product_definition
                IN pd_bag ) ) THEN
                pd_bag := pd_bag +
                change_from_design_object_request_assignment[i].items[j].
                relating_product_definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pd_bag) by 1;
    IF ( NOT (pass1 AND pass2) ) THEN ESCAPE;
    END_IF;
    IF ( pass1 ) THEN
        mdo_bag := [];
        pdr_bag := QUERY( pdr <* USEDIN(pd_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
        (pdr\product_definition_relationship.name = 'design object change') );
        REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
            IF ( NOT pass1 ) THEN ESCAPE;
            END_IF;
            ctdoa_bag := QUERY( ctdoa <* change_to_design_object_assignment |
                (pdr_bag[j] IN ctdoa.items) );
            REPEAT k := 1 to SIZEOF(ctdoa_bag) by 1;
                IF ( NOT pass1 ) THEN ESCAPE;
                END_IF;
                REPEAT l := 1 to SIZEOF(ctdoa_bag[k].items) by 1;
                    IF EXISTS( ctdoa_bag[k].items[l] ) THEN
                        IF ( ctdoa_bag[k].items[l] IN mdo_bag ) THEN
                            pass1 := FALSE;
                            ESCAPE;
                        ELSE
                            mdo_bag := mdo_bag + ctdoa_bag[k].items[l];
                        END_IF;
                    END_IF;
                END_REPEAT;
            END_REPEAT;
        END_REPEAT;
        REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
            IF ( NOT pass1 ) THEN ESCAPE;
            END_IF;

```

```

ctdora_bag :=
  QUERY( ctdora <* change_to_design_object_request_assignment |
        (pdr_bag[j] IN ctdora.items) );
REPEAT k := 1 to SIZEOF(ctdora_bag) by 1;
  IF ( NOT pass1 ) THEN ESCAPE;
  END_IF;
  REPEAT l := 1 to SIZEOF(ctdora_bag[k].items) by 1;
    IF EXISTS( ctdora_bag[k].items[l] ) THEN
      IF ( ctdora_bag[k].items[l] IN mdo_bag ) THEN
        pass1 := FALSE;
        ESCAPE;
      ELSE
        mdo_bag := mdo_bag + ctdora_bag[k].items[l];
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
END_REPEAT;
END_IF;
IF ( pass2 ) THEN
  mdo_bag := [];
  pdr_bag := QUERY( pdr <* USEDIN(pd_bag[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (pdr\product_definition_relationship.name = 'design object change') );
  REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
    IF ( NOT pass2 ) THEN ESCAPE;
    END_IF;
    cfdoa_bag := QUERY( cfdoa <* change_from_design_object_assignment |
                      (pdr_bag[j] IN cfdoa.items) );
    REPEAT k := 1 to SIZEOF(cfdoa_bag) by 1;
      IF ( NOT pass2 ) THEN ESCAPE;
      END_IF;
      REPEAT l := 1 to SIZEOF(cfdoa_bag[k].items) by 1;
        IF EXISTS( cfdoa_bag[k].items[l] ) THEN
          IF ( cfdoa_bag[k].items[l] IN mdo_bag ) THEN
            pass2 := FALSE;
            ESCAPE;
          ELSE
            mdo_bag := mdo_bag + cfdoa_bag[k].items[l];
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
  REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
    IF ( NOT pass2 ) THEN ESCAPE;
    END_IF;
    cfdora_bag :=
      QUERY( cfdora <* change_from_design_object_request_assignment |
            (pdr_bag[j] IN cfdora.items) );
    REPEAT k := 1 to SIZEOF(cfdora_bag) by 1;

```

```

        IF ( NOT pass2 ) THEN ESCAPE;
      END_IF;
      REPEAT l := 1 to SIZEOF(cfdora_bag[k].items) by 1;
        IF EXISTS( cfdora_bag[k].items[l] ) THEN
          IF ( cfdora_bag[k].items[l] IN mdo_bag ) THEN
            pass2 := FALSE;
            ESCAPE;
          ELSE
            mdo_bag := mdo_bag + cfdora_bag[k].items[l];
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
END_IF;
END_REPEAT;
WHERE

    WR1: pass1;

    WR2: pass2;
END_RULE; -- change_design_object_management_relationship_unique_constraint
( *
```

Formal propositions:

WR1: Every instance of `change_to_design_object_assignment` and `change_to_design_object_request_assignment` that has a common `product_definition` that is the related `product_definition` of a `product_definition_relationship` that has a name of 'design object change' where the `product_definition_relationship` is a member of the `items` attribute of the `change_to_design_object_assignment` and `change_to_design_object_request_assignment` must have unique managed_design object entities in their `items` attributes.

WR2: Every instance of `change_from_design_object_assignment` and `change_from_design_object_request_assignment` that has a common `product_definition` that is the relating `product_definition` of a `product_definition_relationship` that has a name of 'design object change' where the `product_definition_relationship` is a member of the `items` attribute of the `change_from_design_object_assignment` and `change_from_design_object_request_assignment` must have unique managed_design object entities in their `items` attributes.

5.2.3.32 change_request_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `change_request_unique_constraint` rule constrains the `versioned_action_request` population members, that fill the role of ARM `change_request`, so that the combination of ARM `version` and ARM `work_request_id` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE change_request_unique_constraint FOR
  ( versioned_action_request );
LOCAL
  cr : BAG OF versioned_action_request :=
    QUERY( v <* versioned_action_request
      | (SIZEOF(QUERY(ara <*
USEDIN(v, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST')|(
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CHANGE_REQUEST' IN TYPEOF(ara)) ))>0) );
  version_bag : BAG OF STRING := [];
  var_bag : BAG OF versioned_action_request;
  pass : BOOLEAN := TRUE;
  id_bag : BAG OF STRING;
END_LOCAL;

REPEAT i := 1 to SIZEOF(versioned_action_request) by 1;
  IF EXISTS( versioned_action_request[i].version ) THEN
    IF( NOT( versioned_action_request[i].version IN version_bag ) ) THEN
      version_bag := version_bag + versioned_action_request[i].version;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(version_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  var_bag := QUERY( v <* versioned_action_request |
    (v\versioned_action_request.version = version_bag[i]) );
  id_bag := [];
  REPEAT j := 1 to SIZEOF(var_bag) by 1;
    IF EXISTS( var_bag[j].id ) THEN
      IF ( var_bag[j].id IN id_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        id_bag := id_bag + var_bag[j].id;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- change_request_unique_constraint
(

```

Formal propositions:

WR1: Every instance of `versioned_action_request` that is the `assigned_action_request` of a `change_request` entity must have a unique combination of `version` and `id` attributes.

5.2.3.33 **characteristic_constraint**

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `characteristic_constraint` rule constrains the use of a `representation_item` or a `measure_representation_item` when it is used as a characteristic.

EXPRESS specification:

```
*)
(*
RULE characteristic_constraint FOR
  (representation_item, measure_representation_item);
END_RULE; -- characteristic_constraint
*)
(*
```

Informal propositions:

IP1: Each `representation_item` that is supporting the ARM concept of a characteristic shall be specified as the item in an `applied_document_reference`.

5.2.3.34 **unique_characterized_product_category**

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `characterized_product_category_unique_constraint` rule constrains the `characterized_product_category` members so that the combination of `name` inherited from `product_category` and the document referenced by `applied_document_reference` `assigned_document` shall be unique within those members.

EXPRESS specification:

```

*)
RULE characterized_product_category_unique_constraint FOR
    (characterized_product_category);
LOCAL
    adr : BAG OF applied_document_reference := [];
    documents : BAG OF document := [];
    labels : BAG OF label := [];
    pass : BOOLEAN := TRUE;
    size_of : BAG OF INTEGER := [];
END_LOCAL;
-- build the set of applied_document_reference that reference characterized_product_cate
-- build the set of labels that characterized_product_category references as name.

REPEAT i := 1 TO SIZEOF(characterized_product_category);
    adr := adr + USEDIN(characterized_product_category[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS');
    labels := labels + characterized_product_category[i]\product_category.name;
END_REPEAT;

IF SIZEOF(adr) <> SIZEOF(labels)
THEN pass := FALSE;
END_IF;

REPEAT k := 1 TO SIZEOF(adr);
    documents := documents + adr[k]\document_reference.assigned_document;
END_REPEAT;

IF SIZEOF(documents) <> SIZEOF(labels)
THEN pass := FALSE;
END_IF;

-- check that each label is related to one and only one instance of document
-- check that each document is related to one and only one label
REPEAT i := 1 to SIZEOF(labels);
    REPEAT j := 1 to SIZEOF(documents);
        REPEAT k := 1 to SIZEOF(adr);
            IF ((labels[i] = adr[k].items[1]\product_category.name) AND
                (documents[j] = adr[k]\document_reference.assigned_document))
            THEN
                size_of[i] := size_of[i] + 1;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 to SIZEOF(labels);
    IF (size_of[i] <> 1)
    THEN pass := FALSE;
    END_IF;

```

```

END_REPEAT;

WHERE

    WR1: pass;
END_RULE; -- via_template_terminal_unique_constraint
( *

```

Formal propositions:

WR1: No more than one instance of `characterized_product_category` shall relate a specific `product_category.name` to a specific document through `applied_document_reference`.

NOTE The same label may be used for different `characterized_product_category` when different documents are referenced. This is the mechanism that permits category names to be isolated in the different contexts established by different dictionaries.

5.2.3.35 `component_external_reference_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `component_external_reference_constraint` rule constrains the use of a representation when it is used as a component external reference.

EXPRESS specification:

```

*)
RULE component_external_reference_constraint FOR
    (representation);
WHERE

    WR1: SIZEOF( QUERY( rep <* representation | (rep\representation.name =
        'component external reference') AND
        (SIZEOF( QUERY( dri <* rep.items | NOT(
            ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(dri)) AND
            (dri\representation_item.name = 'design owner')))) = 1)))=0;

    WR2: SIZEOF(QUERY (rep <* representation |
        (rep\representation.name = 'component external reference') AND
        (SIZEOF (QUERY (dri <* rep.items |
            NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(dri)) AND
            (dri\representation_item.name = 'part number')))) = 1)))=0;

```



```

WR3: sizeof(query (rep <* representation |
  (rep\representation.name = 'component external reference') AND
  (sizeof (query (dri <* rep.items |
    NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN typeof(dri)) AND
    (dri\representation_item.name = 'revision code'))))) = 1)))=0;

WR4: sizeof(query (rep <* representation |
  (rep\representation.name = 'component external reference') AND
  (sizeof (query (dri <* rep.items |
    NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN typeof(dri)) AND
    (dri\representation_item.name = 'product definition id'))))) = 1)))=0;

WR5: sizeof(query (rep <* representation |
  (rep\representation.name = 'component external reference') AND
  (sizeof (query (dri <* rep.items |
    NOT(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN typeof(dri)) AND
    (dri\representation_item.name = 'reference designation'))))) = 1)))=0;
END_RULE; -- component_external_reference_constraint
( *

```

Formal propositions:

WR1: Each representation with a name of 'component external reference' shall contain a descriptive_representation_item with a name of 'design owner'.

WR2: Each representation with a name of 'component external reference' shall contain a descriptive_representation_item with a name of 'part number'.

WR3: Each representation with a name of 'component external reference' shall contain a descriptive_representation_item with a name of 'revision code'.

WR4: Each representation with a name of 'component external reference' shall contain a descriptive_representation_item with a name of 'product definition id'.

WR5: Each representation with a name of 'component external reference' shall contain a descriptive_representation_item with a name of 'reference designation'.

5.2.3.36 component_feature_external_reference_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_feature_external_reference_constraint rule constrains the use of a descriptive_representation_item when it is used as a component feature external reference.

EXPRESS specification:

```

*)
( *
RULE component_feature_external_reference_constraint FOR
  (descriptive_representation_item);
END_RULE; -- component_feature_external_reference_constraint
*)
( *

```

Informal propositions:

IP1: Each descriptive_representation_item with a name of 'component feature external reference' shall be a member of items for a representation with a name of 'component external reference'.

IP2: For each descriptive_representation_item with a name of 'component feature external reference' the combination of the its description and the representation with a name of 'component external reference' of which it is a member of items shall be unique.

5.2.3.37 component_feature_external_reference_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_feature_external_reference_unique_constraint rule constrains the property_definition_-representation population members, that fill the role of ARM component_feature_external_reference, so that the combination of ARM name and ARM associated_component attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_feature_external_reference_unique_constraint FOR
  ( descriptive_representation_item );
LOCAL
  cfer : BAG OF descriptive_representation_item :=
    QUERY( dri <* descriptive_representation_item
| (dri\representation_item.name = 'component feature external reference') );
  desc_bag : BAG OF STRING := [];
  dri_bag : BAG OF descriptive_representation_item;
  rep_bag : BAG OF representation;
  pass : BOOLEAN := TRUE;
  r_bag : BAG OF representation;
END_LOCAL;

REPEAT i := 1 to SIZEOF(cfer) by 1;
  IF EXISTS( cfer[i]\descriptive_representation_item.description ) THEN
    IF ( NOT ( cfer[i]\descriptive_representation_item.description IN desc_bag )) THEN
      desc_bag := desc_bag + cfer[i]\descriptive_representation_item.description;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(desc_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  r_bag := [];
  dri_bag := QUERY( dri <* cfer | (dri\descriptive_representation_item.description = desc_
REPEAT j := 1 to SIZEOF(dri_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  rep_bag := QUERY( r <* USEDIN( dri_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REPRESENTATION.ITEMS' ) | (r\representation.name = 'component external reference' ) );
  REPEAT k := 1 to SIZEOF(rep_bag) by 1;
    IF EXISTS( rep_bag[k] ) THEN
      IF ( rep_bag[k] IN r_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        r_bag := r_bag + rep_bag[k];
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE

```

```

    WR1: pass;
END_RULE; -- component_feature_external_reference_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of `descriptive_representation_item` with a name of 'component feature external reference' must have a unique combination of description attributes and representation entities where the representations have the `descriptive_representation_items` in their `items` attribute and the representations have a name of 'component external reference'.

5.2.3.38 **component_feature_to_layout_feature_relationship_constraint**

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `component_feature_to_layout_feature_relationship_constraint` rule constrains the related and relating `shape_aspect` of a `shape_aspect_relationship` that is used as a component feature to layout feature.

EXPRESS specification:

```

*)
RULE component_feature_to_layout_feature_relationship_constraint FOR
( shape_aspect_relationship );
WHERE

    WR1: SIZEOF(QUERY(sar <* shape_aspect_relationship |
        (sar\shape_aspect_relationship.name = 'component feature to layout feature')
        AND NOT(SIZEOF(
            [ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'COMPONENT_SHAPE_ASPECT',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'COMPONENT_INTERFACE_TERMINAL',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'COMPONENT_TERMINAL' ]
            * TYPEOF(sar.relying_shape_aspect))) > 0)
        )) = 0;

```

```

WR2: SIZEOF(QUERY(sar <* shape_aspect_relationship |
    (sar\shape_aspect_relationship.name = 'component feature to layout feature')
    AND NOT(SIZEOF(
        [ 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'STRATUM_FEATURE']
        * TYPEOF(sar.related_shape_aspect))) > 0)
    )) = 0;
END_RULE; -- component_feature_to_layout_feature_relationship_constraint
(*

```

Formal propositions:

WR1: IF shape_aspect_relationship.name = 'component feature to layout feature' the shape_aspect_relationship.relate_shape_aspect is a (component_shape_aspect) (component_interface_terminal) (component_terminal).

WR2: IF shape_aspect_relationship.name = 'component feature to layout feature' the shape_aspect_relationship.related_shape_aspect is a (component_shape_aspect) (stratum_feature).

5.2.3.39 component_feature_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_feature_unique_constraint rule constrains the component_terminal and component_interface_terminal population members, that fill the role of ARM component_feature, so that the combination of ARM definition and ARM associated_component attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_feature_unique_constraint FOR
  ( component_terminal, component_interface_terminal );
LOCAL
  ct : BAG OF shape_aspect :=
    QUERY( ct <* component_terminal | (ct\shape_aspect.description IN
['assembly module component terminal',
'bare die component terminal',
'interconnect component join terminal',
'interconnect module component terminal',
'interconnect module component stratum based terminal',
'minimally defined component terminal',
'packaged component join terminal']) );
  cit : BAG OF shape_aspect :=
    QUERY( cit <* component_interface_terminal | (
cit\shape_aspect.description IN ['interconnect component interface terminal',
'packaged connector component interface terminal']) );
  cf : BAG OF shape_aspect;
  cd_bag : BAG OF component_definition := [];
  sa_bag : BAG OF shape_aspect;
  sar_bag : BAG OF shape_aspect_relationship;
  pass : BOOLEAN := TRUE;
  def_bag : BAG OF shape_aspect;
END_LOCAL;

cf := ct + cit;
REPEAT i := 1 to SIZEOF(cf) by 1;
  IF EXISTS( cf[i].of_shape.definition ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION' IN TYPEOF(cf[i].of_shape.definition) ) THEN
      IF ( NOT ( cf[i].of_shape.definition IN cd_bag )) THEN
        cd_bag := cd_bag + cf[i].of_shape.definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(cd_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  def_bag := [];
  sa_bag := QUERY( sa <* cf | (sa.of_shape.definition :=: cd_bag[i]) );
  REPEAT j := 1 to SIZEOF(sa_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    sar_bag := QUERY( sar <* USEDIN( sa_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT' ) |
      (sar\shape_aspect_relationship.name = 'instantiated feature') );
  END_REPEAT;
END_REPEAT;

```

```

REPEAT k := 1 to SIZEOF(sar_bag) by 1;
  IF EXISTS( sar_bag[k].relating_shape_aspect ) THEN
    IF ( sar_bag[k].relating_shape_aspect IN def_bag ) THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      def_bag := def_bag + sar_bag[k].relating_shape_aspect;
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_feature_unique_constraint
(*)

```

Formal propositions:

WR1: Every instance of component_terminal with a name of 'assembly module component terminal', 'bare die component terminal', 'interconnect component join terminal', 'interconnect module component terminal', 'interconnect module component stratum based terminal', 'minimally defined component terminal' or 'packaged component join terminal' and component_interface_terminal with a name of either 'interconnect component interface terminal' or 'packaged connector component interface terminal' must have a unique combination of of_shape.definition attributes and shape_aspect entities where the shape_aspects are related to the component_terminals and component_interface_terminals by a shape_aspect_relationship with a name of 'instantiated feature'.

5.2.3.40 component_mounting_clearance_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_mounting_clearance_relationship_constraint rule constrain the items of a representation and constrain the usage of that representation when it is used as a component mounting clearance.

EXPRESS specification:

```

*)
RULE component_mounting_clearance_relationship_constraint FOR
( representation );
WHERE

```

```

WR1: SIZEOF(QUERY(rep <* representation |
  (rep\representation.name = 'mounting clearance')
  AND (SIZEOF(QUERY(lmwu <* rep.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(lmwu))
    AND (lmwu\representation_item.name = 'minimum height')
  )) = 0))) = 0;
END_RULE; -- component_mounting_clearance_relationship_constraint
( *

```

Formal propositions:

WR1: IF the representation name = 'mounting clearance' it shall contain one item that is a length_measure_with_unit that has a name = 'minimum height'.

Informal propositions:

IP1: IF the representation name = 'mounting clearance' there shall be one property_definition_representation for which the representation is the used_representation and the definition for the property_definition_representation is a property_definition with a name equal to 'component mounting property', and the shape_aspect for the property_definition is a component_feature_relationship.

5.2.3.41 component_overlap_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_overlap_relationship_constraint rule constrain the items of a representation and constrain the usage of that representation when it is used as a component overlap clearance.

EXPRESS specification:

```

*)
RULE component_overlap_relationship_constraint FOR
( representation );
WHERE

WR1: SIZEOF(QUERY(rep <* representation |
  (rep\representation.name = 'overlap clearance')
  AND (SIZEOF(QUERY(lmwu <* rep.items |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(lmwu))
    AND (lmwu\representation_item.name = 'minimum clearance')
  )) = 0))) = 0;
END_RULE; -- component_overlap_relationship_constraint
( *

```


Formal propositions:

WR1: IF the representation name = 'overlap clearance' it shall contain one item that is a length_measure_with_unit that has a name = 'minimum clearance'.

Informal propositions:

IP1: IF the representation name = 'overlap clearance' there shall be two property_definition_representations for which the representation is the used_representation and the definition for the property_definition_representations is a property_definition with a name equal to 'component mounting property' and one of the property_definitions has a description = 'current component' and the second property_definition has a description = 'previously placed component'.

IP2: IF the representation name = 'overlap clearance' the two property_definition_representations for which the representation is the used_representation must have two product_definition_relationships which point to one product_definition.

5.2.3.42 component_part_2d_geometric_representation_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_part_2d_geometric_representation_relationship_constraint rule constrains the rep_1 and rep_2 attributes of a shape_representation_relationship when it is used to relate a component planar shape and the defining part planar shape.

EXPRESS specification:

```

*)
RULE component_part_2d_geometric_representation_relationship_constraint FOR
( shape_representation_relationship );
LOCAL
  rr: SET OF representation_relationship :=
    QUERY ( srr <* shape_representation_relationship |
            srr\representation_relationship.name =
              'component part planar shape' );
  pass: BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 to SIZEOF(rr) by 1;
IF ((rr[i].rep_1.context_of_items.coordinate_space_dimension) <>
    (rr[i].rep_2.context_of_items.coordinate_space_dimension))
THEN
  pass := FALSE;
END_IF;
END_REPEAT;
WHERE

```

```

WR1: pass;
END_RULE; -- component_part_2d_geometric_-
          -- representation_relationship_constraint
( *

```

Formal propositions:

WR1: IF shape_representation_relationship\representation_relationship.name = 'component part planar shape' the representation_relationship.rep_1 shall have the same dimensionality as representation_relationship.rep_2.

Informal propositions:

IP1: IF shape_representation_relationship\representation_relationship.name = 'component part planar shape' the representation_relationship.rep_1 shall be a shape_representation with name = 'planar projected shape' that is the used_representation of a property_definition_representation where property_definition_representation references an instance through the definition attribute that (is a physical_unit) (is a part_template_definition whose shape is defined by a characterized_object).

IP2: IF shape_representation_relationship\representation_relationship.name = 'component part planar shape' the representation_relationship.rep_2 shall be a shape_representation with name = 'planar projected shape' that is the used_representation of a property_definition_representation where ({property_definition_representation definition is a product_definition_shape} {property_definition_representation references a product_definition that is a component_definition}) (property_definition_representation references a shape_aspect that is a component_shape_aspect).

IP3: The combination of representation_relationship.rep_1, representation_relationship.rep_2 shall be unique within a population of representation_relationship with a representation_relationship name = 'component part planar shape'.

5.2.3.43 component_part_3d_geometric_representation_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_part_3d_geometric_representation_relationship_constraint rule constrains the rep_1 and rep_2 attributes of a shape_representation_relationship when it is used to relate a component 3d shape and the defining part 3d shape.

EXPRESS specification:

```

*)
RULE component_part_3d_geometric_representation_relationship_constraint FOR
( shape_representation_relationship );
LOCAL
  rr: SET OF representation_relationship :=
    QUERY ( srr <* shape_representation_relationship |
      srr\representation_relationship.name = 'component part 3d shape' );
  pass: BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 to SIZEOF(rr) by 1;
IF ((rr[i].rep_1.context_of_items.coordinate_space_dimension) <>
    (rr[i].rep_2.context_of_items.coordinate_space_dimension))
THEN
  pass := FALSE;
END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_part_3d_geometric_representation_-
          -- relationship_constraint
( *

```

Formal propositions:

WR1: IF shape_representation_relationship\representation_relationship.name = 'component part 3d shape' the representation_relationship.rep_1 shall have the same context dimensionality as representation_relationship.rep_2.

Informal propositions:

IP1: IF shape_representation_relationship\representation_relationship.name = 'component part 3d shape' the representation_relationship.rep_1 shall be a shape_representation with name = '3d bound volume shape' or 'part template non planar 2d shape' that is the used_representation of a property_definition_representation where property_definition_representation.definition.definition (is a physical_unit) (is a part_template_definition whose shape is defined by a characterized_object).

IP2: IF shape_representation_relationship\representation_relationship.name = 'component part 3d shape' the representation_relationship.rep_2 shall be a shape_representation with name = '3d bound volume shape' that is the used_representation of a property_definition_representation where ({property_definition_representation definition is a product_definition_shape} {property_definition_representation definition references a component_definition}) (property_definition_representation definition references a component_shape_aspect).

IP3: The combination of representation_relationship.rep1 and representation_relationship.rep2 shall be unique.

5.2.3.44 component_placement_restriction_assignment_constraint

The component_placement_restriction_assignment_constraint rule constrains a representation that is used as a component placement restriction assignment. This rule implements WR1 of the Application object.

EXPRESS specification:

```

*)
RULE component_placement_restriction_assignment_constraint FOR
( representation );
LOCAL
  cpra : SET OF representation := QUERY( cpra <* representation |
    cpra.name = 'component placement restriction assignment');
  i : INTEGER ;
  pass1 : BOOLEAN := TRUE;
  pass2 : BOOLEAN := TRUE;
  mra : BOOLEAN := FALSE;
  mrv : BOOLEAN := FALSE;
END_LOCAL;
REPEAT i := 1 to SIZEOF (cpra);
  IF NOT(pass1) THEN ESCAPE; END_IF;

  IF NOT((SIZEOF(QUERY(rr <* USEDIN(cpra[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name =
    'component placement restriction assignment area') AND
    (SIZEOF(QUERY(pdr <* USEDIN(rr.rep_2,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_AREA' IN TYPEOF(pdr.definition.definition)))
    ) = 1 )
    )) <=1 ) OR
  (SIZEOF(QUERY(rr <* USEDIN(cpra[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name =
    'component placement restriction assignment volume') AND
    (SIZEOF(QUERY(pdr <* USEDIN(rr.rep_2,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_VOLUME' IN TYPEOF(pdr.definition.definition)))
    ) = 1 )
    )) <=1 )
    ) THEN pass1 := FALSE;
  END_IF;
  mra := (SIZEOF(QUERY(rr <* USEDIN(cpra[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name =
    'component placement restriction assignment area') AND
    (SIZEOF(QUERY(pdr <* USEDIN(rr.rep_2,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_AREA' IN TYPEOF(pdr.definition.definition)))
    ) = 1 )
    )) <=1 )

```

```

    )) =1 );
    mrv := (SIZEOF(QUERY(rr <* USEDIN(cpra[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name =
    'component placement restriction assignment volume') AND
    (SIZEOF(QUERY(pdr <* USEDIN(rr.rep_2,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_VOLUME' IN TYPEOF(pdr.definition.definition)))
    ) = 1 )
    )) =1 );
    IF NOT(mra OR mrv) THEN pass1 := FALSE; END_IF;
END_REPEAT;
REPEAT i := 1 to SIZEOF (cpra);
    IF NOT(pass1) THEN ESCAPE; END_IF;
    IF NOT(pass2) THEN ESCAPE; END_IF;

    IF NOT(SIZEOF(QUERY(pdr <* USEDIN(cpra[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUPED_REQUIREMENTS_PROPERTY' IN TYPEOF(pdr.definition))
    AND
    (pdr.definition\group.name =
    'item restricted requirements property')
    )
    ) = 1 ) THEN pass2 := FALSE;
    END_IF;
END_REPEAT;
WHERE

    WR1: pass1;

    WR2: pass2;
END_RULE; -- component_placement_restriction_assignment_constraint
( *
```

Formal propositions:

WR1: IF representation_relationship.name = 'component placement restriction assignment area' the shape_aspect at the end of the path starting with rep_2 through used_representation of a property_definition_representation and ending with property_definition_representation.definition.definition shall be a mounting_restriction_area.

IF representation_relationship.name = 'component placement restriction assignment volume' the shape_aspect at the end of the path starting with rep_2 through used_representation of a property_definition_representation and ending with property_definition_representation.definition.definition shall be a mounting_restriction_volume.

At least one of area or volume shall exist.

WR2: IF representation.name = 'component placement restriction assignment' the representation shall be referenced as a used_representation by a property_definition_representation that shall have a definition that shall be a grouped_requirements_property which has a name = 'item restricted requirements property'.

5.2.3.45**component_terminal_to_assembly_module_interface_terminal_assignment_constraint**

The component_terminal_to_assembly_module_interface_terminal_assignment_constraint rule constrains the related_shape_aspect of a shape_aspect_relationship when it is used as a component terminal to assembly module terminal assignment.

EXPRESS specification:

```
* )
( *
RULE component_terminal_to_assembly_module_interface_terminal_assignment_constraint
  FOR ( shape_aspect_relationship );
END_RULE; -- component_terminal_to_assembly_module_interface_terminal_
          -- assignment_constraint
* )
( *
```

Informal propositions:

IP1: IF {shape_aspect_relationship.name = 'component feature to physical usage view assignment'} {shape_aspect_relationship.relate_shape_aspect is an assembly_module_interface_terminal where the product_definition referenced through the path assembly_module_interface_terminal to it's supertype shape_aspect.of_shape.definition.name = 'assembly module'} the shape_aspect_relationship.related_shape_aspect shall be a (component_terminal where component_terminal\shape_aspect.description = ('assembly module component terminal') ('interconnect module component terminal')) (component_interface_terminal)

5.2.3.46 component_terminal_to_assembly_module_interface_terminal_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_assembly_module_interface_terminal_assignment_unique_constraint rule constrains the shape_aspect_relationship population members, that fill the role of ARM component_terminal_to_assembly_module_interface_terminal_assignment, so that the ARM usage_view_terminal attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE
component_terminal_to_assembly_module_interface_terminal_assignment_unique_constraint
FOR
  ( shape_aspect_relationship );
LOCAL
  cttamta : BAG OF shape_aspect_relationship := QUERY( sar <*
shape_aspect_relationship | (sar\shape_aspect_relationship.name =
      'component feature to physical usage view assignment') );
  pass : BOOLEAN := TRUE;
  amt_bag : BAG OF assembly_module_interface_terminal := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cttamta) by 1;
  IF EXISTS( cttamta[i].relating_shape_aspect ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_INTERFACE_TERMINAL' IN TYPEOF(cttamta[i].relating_shape_aspect) )
    THEN
      IF ( cttamta[i].relating_shape_aspect IN amt_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        amt_bag := amt_bag + cttamta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_terminal_to_assembly_module_interface_terminal_assignment_-
          -- unique_constraint
( *
```


Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'component feature to physical usage view assignment' must have a unique assembly_module_interface_terminal entity as the shape_aspect_relationship's relating_shape_aspect attribute.

5.2.3.47**component_terminal_to_assembly_module_join_terminal_assignment_constraint**

The component_terminal_to_assembly_module_join_terminal_assignment_constraint rule constrains the related_shape_aspect of a shape_aspect_relationship when it is used as a component terminal to assembly module terminal assignment.

EXPRESS specification:

```

*)
( *
RULE component_terminal_to_assembly_module_join_terminal_assignment_constraint
  FOR ( shape_aspect_relationship );
END_RULE; -- component_terminal_to_assembly_module_join_terminal_
          -- assignment_constraint
*)
( *

```

Informal propositions:

IP1: IF {shape_aspect_relationship.name = 'component feature to physical usage view assignment'} {shape_aspect_relationship.relying_shape_aspect is an assembly_module_join_terminal where the product referenced through the path assembly_module_join_terminal to it's supertype shape_aspect.of_shape.definition is a product_definition has a name = 'assembly module'} the shape_aspect_relationship.related_shape_aspect shall be a (component_terminal where component_terminal.shape_aspect.description = ('assembly module component terminal') ('interconnect module component terminal')) (component_interface_terminal)

5.2.3.48 component_terminal_to_assembly_module_join_terminal_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_assembly_module_join_terminal_assignment_unique_constraint rule constrains the shape_aspect_relationship population members, that fill the role of ARM component_terminal_to_assembly_module_join_terminal_assignment, so that the ARM usage_view_terminal attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE
component_terminal_to_assembly_module_join_terminal_assignment_unique_constraint
FOR
  ( shape_aspect_relationship );
LOCAL
  cttamta : BAG OF shape_aspect_relationship := QUERY( sar <*
shape_aspect_relationship | (sar\shape_aspect_relationship.name =
    'component feature to physical usage view assignment') );
  pass : BOOLEAN := TRUE;
  amt_bag : BAG OF assembly_module_join_terminal := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cttamta) by 1;
  IF EXISTS( cttamta[i].relating_shape_aspect ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_JOIN_TERMINAL' IN TYPEOF(cttamta[i].relating_shape_aspect) )
    THEN
      IF ( cttamta[i].relating_shape_aspect IN amt_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        amt_bag := amt_bag + cttamta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_terminal_to_assembly_module_join_terminal_assignment_-
          -- unique_constraint
( *

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'component feature to physical usage view assignment' must have a unique assembly_module_join_terminal entity as the shape_aspect_relationship's relating_shape_aspect attribute.

5.2.3.49 component_terminal_to_interconnect_module_interface_terminal_assignment_constraint

The component_terminal_to_interconnect_module_interface_terminal_assignment_constraint rule constrains the related_shape_aspect of a shape_aspect_relationship when it is used as a component terminal to interconnect module terminal assignment.

EXPRESS specification:

```

*)
( *
RULE
  component_terminal_to_interconnect_module_interface_terminal_assignment_constraint
FOR ( shape_aspect_relationship );
END_RULE; -- component_terminal_to_assembly_module_interface_terminal_assignment_-
          -- constraint
*)
( *

```

Informal propositions:

IP1: IF {shape_aspect_relationship.name = 'component feature to physical usage view assignment'} {shape_aspect_relationship.relate_shape_aspect is an assembly_module_interface_terminal where the product referenced through the path assembly_module_interface_terminal to its supertype shape_aspect.of_shape.definition is a product_definition that has a name = 'interconnect module'} the shape_aspect_relationship.related_shape_aspect shall be a (component_shape_aspect where component_shape_aspect.description = ('component termination passage interface terminal') ('land interface terminal') ('non functional land interface terminal') ('printed connector component interface terminal')).

5.2.3.50 component_terminal_to_interconnect_module_interface_terminal_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_interconnect_module_interface_terminal_assignment_unique_constraint rule constrains the shape_aspect_relationship population members, that fill the role of ARM component_terminal_to_interconnect_module_interface_terminal_assignment, so that the ARM usage_view_terminal attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_terminal_to_interconnect_module_interface_terminal_assignment_unique_constraint
FOR
  ( shape_aspect_relationship );
LOCAL
  cttimta : BAG OF shape_aspect_relationship := QUERY( sar <*
shape_aspect_relationship | (sar\shape_aspect_relationship.name =
  'component feature to physical usage view assignment') );
  pass : BOOLEAN := TRUE;
  imt_bag : BAG OF interconnect_module_interface_terminal := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cttimta) by 1;
  IF EXISTS( cttimta[i].relating_shape_aspect ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_INTERFACE_TERMINAL' IN
      TYPEOF(cttimta[i].relating_shape_aspect) ) THEN
      IF ( cttimta[i].relating_shape_aspect IN imt_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        imt_bag := imt_bag + cttimta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_terminal_to_interconnect_module_interface_terminal_assignment_unique_constraint
(*)

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'component feature to physical usage view assignment' must have a unique interconnect_module_interface_terminal entity as the shape_aspect_relationship's relating_shape_aspect attribute.

5.2.3.51**component_terminal_to_interconnect_module_join_terminal_assignment_constraint**

The component_terminal_to_interconnect_module_join_terminal_assignment_constraint rule constrains the related_shape_aspect of a shape_aspect_relationship when it is used as a component terminal to interconnect module terminal assignment.

EXPRESS specification:

```

*)
( *
RULE
  component_terminal_to_interconnect_module_join_terminal_assignment_constraint
  FOR ( shape_aspect_relationship );
END_RULE; -- component_terminal_to_assembly_module_join_terminal_assignment_-
          -- constraint
*)
( *

```

Informal propositions:

IP1: IF {shape_aspect_relationship.name = 'component feature to physical usage view assignment'} {shape_aspect_relationship.relate_shape_aspect is an assembly_module_join_terminal where the product referenced through the path assembly_module_join_terminal to its supertype shape_aspect.of_shape.definition is a product_definition that has a name = 'interconnect module'} the shape_aspect_relationship.related_shape_aspect shall be a (component_shape_aspect where component_shape_aspect.description = ('component termination passage interface terminal') ('land interface terminal') ('non functional land interface terminal') ('printed connector component interface terminal')).

5.2.3.52 component_terminal_to_interconnect_module_join_terminal_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_interconnect_module_join_terminal_assignment_unique_constraint rule constrains the shape_aspect_relationship population members, that fill the role of ARM component_terminal_to_interconnect_module_join_terminal_assignment, so that the ARM usage_view_terminal attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_terminal_to_interconnect_module_join_terminal_assignment_unique_constraint
FOR
  ( shape_aspect_relationship );
LOCAL
  cttimta : BAG OF shape_aspect_relationship := QUERY( sar <*
shape_aspect_relationship | (sar\shape_aspect_relationship.name =
  'component feature to physical usage view assignment') );
  pass : BOOLEAN := TRUE;
  imt_bag : BAG OF interconnect_module_join_terminal := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cttimta) by 1;
  IF EXISTS( cttimta[i].relating_shape_aspect ) THEN
    IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_JOIN_TERMINAL' IN
      TYPEOF(cttimta[i].relating_shape_aspect) ) THEN
      IF ( cttimta[i].relating_shape_aspect IN imt_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        imt_bag := imt_bag + cttimta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_terminal_to_interconnect_module_join_terminal_assignment_unique_con
(

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'component feature to physical usage view assignment' must have a unique interconnect_module_join_terminal entity as the shape_aspect_relationship's relating_shape_aspect attribute.

5.2.3.53 component_terminal_to_pca_terminal_assignment_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_pca_terminal_assignment_constraint rule constrains the related_shape_aspect of a shape_aspect_relationship when it is used as a component terminal to pca terminal assignment.

EXPRESS specification:

```

*)
( *
RULE component_terminal_to_pca_terminal_assignment_constraint FOR
( shape_aspect_relationship );
END_RULE; -- component_terminal_to_pca_terminal_assignment_constraint
*)
( *

```

Informal propositions:

IP1: IF {shape_aspect_relationship.name = 'component feature to physical usage view assignment'} {shape_aspect_relationship.relying_shape_aspect is an assembly_module_terminal\shape_aspect.description = 'pca terminal'} the shape_aspect_relationship.related_shape_aspect shall be a (component_terminal where component_terminal\shape_aspect.description = ('assembly module component terminal') ('interconnect module component terminal')) (component_interface_terminal) (assembly_module_terminal where assembly_module_terminal\shape_aspect.description = 'pca terminal').

5.2.3.54 component_terminal_to_pca_terminal_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_terminal_to_pca_terminal_assignment_unique_constraint rule constrains the shape_aspect_relationship population members, that fill the role of ARM component_terminal_to_pca_terminal_assignment, so that the ARM usage_view_terminal attribute shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_terminal_to_pca_terminal_assignment_unique_constraint FOR
  ( shape_aspect_relationship );
LOCAL
  cttamta : BAG OF shape_aspect_relationship := QUERY( sar <*
shape_aspect_relationship | (sar\shape_aspect_relationship.name =
    'component feature to physical usage view assignment') );
  pass : BOOLEAN := TRUE;
  amt_bag : BAG OF assembly_module_terminal := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cttamta) by 1;
  IF EXISTS( cttamta[i].relating_shape_aspect ) THEN
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT' IN TYPEOF(cttamta[i].relating_shape_aspect)) AND
(cttamta[i].relating_shape_aspect\shape_aspect.description = 'pca terminal') ) THEN
      IF ( cttamta[i].relating_shape_aspect IN amt_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        amt_bag := amt_bag + cttamta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- component_terminal_to_pca_terminal_assignment_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'component feature to physical usage view assignment' must have a unique assembly_module_terminal entity with a description of 'pca terminal' as the shape_aspect_relationship's relating_shape_aspect attribute.

5.2.3.55 component_termination_passage_template_terminal_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_termination_passage_template_terminal_constraint rule constrains the use of a shape_aspect when it is used as a component termination passage template terminal.

EXPRESS specification:

```

*)
( *
RULE component_termination_passage_template_terminal_constraint FOR
( shape_aspect );
END_RULE; -- component_termination_passage_template_terminal_constraint
*)
( *

```

Informal propositions:

IP1: IF a shape_aspect {has a description = ('component termination passage interface terminal') ('component termination passage join terminal')} {is the related_shape_aspect of a shape_aspect_relationship which {has a name = 'access mechanism'} {has a relating_shape_aspect which is a part_template_definition with {a description = 'component termination passage template'} {a of_shape.definition which is a characterized_object}}}} the shape_aspect must be the relating_shape_aspect of at least one shape_aspect_relationship where {shape_aspect_relationship.name = 'terminal connection zone'} {shape_aspect_relationship.related_shape_aspect is a shape_aspect.description = 'connection zone'}.

5.2.3.56 component_termination_passage_template_terminal_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The component_termination_passage_template_terminal_unique_constraint rule constrains the shape_aspect population members, that fill the role of ARM component_termination_passage_template_terminal, so the combination of ARM associated_definition and ARM name shall be unique within the population.

EXPRESS specification:

```

*)
RULE component_termination_passage_template_terminal_unique_constraint FOR
  ( shape_aspect );
LOCAL
  ctptt : BAG OF shape_aspect := QUERY( sa <* shape_aspect |
  ((sa\shape_aspect.description =
  'component termination passage template interface terminal') OR
  (sa\shape_aspect.description =
  'component termination passage template join terminal')) );
  name_bag : BAG OF STRING := [];
  sa_bag : BAG OF shape_aspect;
  sar_bag : BAG OF shape_aspect_relationship;
  pass : BOOLEAN := TRUE;
  ptd_bag : BAG OF part_template_definition;
END_LOCAL;

REPEAT i := 1 to SIZEOF(ctptt) by 1;
  IF EXISTS( ctptt[i]\shape_aspect.name ) THEN
    IF ( NOT ( ctptt[i]\shape_aspect.name IN name_bag ) ) THEN
      name_bag := name_bag + ctptt[i]\shape_aspect.name;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(name_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  ptd_bag := [];
  sa_bag := QUERY( sa <* ctptt | (sa\shape_aspect.name = name_bag[i]) );
  REPEAT j := 1 to SIZEOF(sa_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    sar_bag := QUERY( sar <* USEDIN(sa_bag[j],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    ((sar\shape_aspect_relationship.name = 'access mechanism') AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION' IN TYPEOF(sar.relying_shape_aspect)) AND
    (sar.relying_shape_aspect\shape_aspect.name =
    'component termination passage template')) );
    REPEAT k := 1 to SIZEOF(sar_bag) by 1;
      IF EXISTS( sar_bag[k].relying_shape_aspect ) THEN
        IF ( sar_bag[k].relying_shape_aspect IN ptd_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          ptd_bag := ptd_bag + sar_bag[k].relying_shape_aspect;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

```

```

        END_REPEAT;
    END_REPEAT;
END_REPEAT;
WHERE

    WR1: pass;
END_RULE; -- component_termination_passage_template_terminal_-
        -- unique_constraint
( *
```

Formal propositions:

WR1: Every instance of shape_aspect with a description of either 'component termination passage template interface terminal' or 'component termination passage template join terminal' must have a unique combination of name attribute and part_template_definition entity with a name of 'component termination passage template' where the part_template_definition is related to the shape_aspect by a shape_aspect_relationship with a name of 'access mechanism'.

5.2.3.57 composite_shape_element_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The composite_shape_element_constraint rule constrains the use of a composite_shape_aspect when it is used as a composite shape element or as an implementation of the ARM object Part_group_feature or as an implementation of the ARM object Composite_shape_element.

EXPRESS specification:

```

*)
RULE composite_shape_element_constraint FOR
    (composite_shape_aspect);
WHERE

    WR1: SIZEOF(QUERY(csa <* composite_shape_aspect |
        SIZEOF(QUERY(sar <* USEDIN(csa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
        (sar\shape_aspect_relationship.name = 'composing')
        )) < 2)) = 0;
```

```

WR2: sizeof(QUERY(csa <* composite_shape_aspect |
    NOT ((sizeof(TYPEOF(csa)) > 2) OR
    (csa\shape_aspect.description <> '') OR
    (csa\shape_aspect.name <> ''))
    )) = 0;
END_RULE; -- composite_shape_element_constraint
( *

```

Formal propositions:

WR1: Each composite_shape_aspect shall be specified as the relating_shape_aspect by at least two shape_aspect_relationship that have a name of 'composing'.

WR2: Each composite_shape_aspect shall be at least one other type of shape_aspect.

Informal propositions:

IP1: Each composite_shape_aspect that has a description of 'part group feature' shall only be related to other part features by the 'composing' shape_aspect_relationship.

5.2.3.58 composite_signal_property_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The composite_signal_property_constraint rule constrains the use of a property_definition when it is used as a composite signal property.

EXPRESS specification:

```

*)
( *
RULE composite_signal_property_constraint FOR
( property_definition );
END_RULE; -- composite_signal_property_constraint
*)
( *

```

Informal propositions:

IP1: A property_definition with a name = 'composite signal property' shall be the relating_property_definition of at least one property_definition_relationship with a name = 'composite signal property relationship'.

5.2.3.59 conductive_interconnect_element_terminal_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conductive_interconnect_element_terminal_unique_constraint` rule constrains `component_terminal` population members, that fill the role of ARM `conductive_interconnect_element_terminal`, so that the combination of ARM name and ARM `associated_component` shall be unique within the population.

EXPRESS specification:

```

*)
RULE conductive_interconnect_element_terminal_unique_constraint FOR
  ( component_terminal );
LOCAL
  ciet : BAG OF component_terminal := QUERY( ct <* component_terminal |
(ct\shape_aspect.description = 'conductive interconnect element terminal') );
  name_bag : BAG OF STRING := [];
  ct_bag : BAG OF component_terminal;
  sar_bag : BAG OF shape_aspect_relationship;
  pass : BOOLEAN := TRUE;
  csa_bag : BAG OF component_shape_aspect;
END_LOCAL;

REPEAT i := 1 to SIZEOF(ciet) by 1;
  IF EXISTS( ciet[i]\shape_aspect.name ) THEN
    IF( NOT( ciet[i]\shape_aspect.name IN name_bag ) ) THEN
      name_bag := name_bag + ciet[i]\shape_aspect.name;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(name_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  ct_bag := QUERY( ct <* ciet | (ct\shape_aspect.name = name_bag[i]) );
  csa_bag := [];
  REPEAT j := 1 to SIZEOF(ct_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    sar_bag := QUERY( sar <* USEDIN( ct_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar\shape_aspect_relationship.name =
'associated component') AND (sar.relate_shape_aspect\shape_aspect.description IN
['conductive interconnect element with pre defined transitions',
'conductive interconnect element with user defined single transition'])
AND ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT' IN TYPEOF(sar.relate_shape_aspect))) );
    REPEAT k := 1 to SIZEOF(sar_bag) by 1;
      IF EXISTS( sar_bag[k].relate_shape_aspect ) THEN
        IF( sar_bag[k].relate_shape_aspect IN csa_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          csa_bag := csa_bag + sar_bag[k].relate_shape_aspect;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

```

WHERE

```

    WR1: pass;
END_RULE; -- conductive_interconnect_element_terminal_unique_constraint
( *
```

Formal propositions:

WR1: Every instance of `component_terminal` with a name of 'conductive interconnect element terminal' must have a unique combination of name attributes and `component_shape_aspect` entities that are related via a `shape_aspect_relationship` with a name of 'associated component'.

5.2.3.60 conductive_interconnect_element_with_pre_defined_transitions_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conductive_interconnect_element_with_pre_defined_transitions_constraint` rule constrains the use of a `shape_aspect` when it is used as a conductive interconnect element with pre defined transitions.

EXPRESS specification:

```

*)
( *
RULE
conductive_interconnect_element_with_pre_defined_transitions_constraint
FOR ( shape_aspect );
END_RULE; -- conductive_interconnect_element_with_pre_defined_transitions_
          -- constraint
*)
( *
```

Informal propositions:

IP1: Each `shape_aspect` with {name = 'conductive interconnect element with pre defined transitions'} {of_shape.definition is a component_definition} shall be the definition of exactly one `property_definition_representation` that has a `used_representation` (that is a `shape_representation.name` = 'bound face' and one item is a `face_surface`) (that is a `representation.name` = 'styled curve' and one item is an `annotation_curve_occurrence`).

IP2: Each `shape_aspect` with {name = 'conductive interconnect element with pre defined transitions'} shall be the `relating_shape_aspect` of at least one `shape_aspect_relationship` with a name = 'connected points' where the `related_shape_aspect` is a `layer_connection_point_link`.

IP3: Each shape_aspect with {name = 'conductive interconnect element with pre defined transitions'} {of_shape.definition is a component_definition} shall be the related_shape_aspect of exactly one shape_aspect_relationship with a name = 'instantiated feature' that has a relating_shape_aspect that is a part_template_definition with a description = 'trace template'.

5.2.3.61

conductive_interconnect_element_with_user_defined_single_transition_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The conductive_interconnect_element_with_user_defined_single_transition_constraint rule constrains the use of a shape_aspect when it is used as a conductive interconnect element with user defined single transition.

EXPRESS specification:

```
* )
( *
RULE conductive_interconnect_element_with_user_defined_single_transition_constraint
FOR ( shape_aspect );
END_RULE; -- conductive_interconnect_element_with_user_defined_single_transition_constraint
* )
( *
```

Informal propositions:

IP1: Each shape_aspect with {name = 'conductive interconnect element with user defined single transition'} {of_shape.definition is a component_definition} shall be the relating_shape_aspect of one shape_aspect_relationship where {shape_aspect_relationship.name = 'end terminus'} {shape_aspect_relationship.related_shape_aspect is a layer_connection_point}.

IP2: Each shape_aspect with {name = 'conductive interconnect element with user defined single transition'} {of_shape.definition is a component_definition} shall be the relating_shape_aspect of one shape_aspect_relationship where {shape_aspect_relationship.name = 'start terminus'} {shape_aspect_relationship.related_shape_aspect is a layer_connection_point}.

IP3: Each shape_aspect with {name = 'conductive interconnect element with user defined single transition'} {of_shape.definition is a component_definition} shall participate as a definition of exactly two property_definition_representations whose used_representation is a shape_representation and one shape_representation.items is a curve.

IP4: Each shape_aspect with {name = 'conductive interconnect element with user defined single transition'} {of_shape.definition is a component_definition} shall be the related_shape_aspect of one shape_aspect_relationship where {shape_aspect_relationship.name = 'instantiated feature'} {shape_aspect_relationship.description = 'end feature'} {shape_aspect_relationship.relying_shape_aspect is a part_template_definition whose description = 'trace template'}.

IP5: Each shape_aspect with {name = 'conductive interconnect element with user defined single transition'} {of_shape.definition is a component_definition} shall be the related_shape_aspect of one shape_aspect_relationship where {shape_aspect_relationship.name = 'instantiated feature'} {shape_aspect_relationship.description = 'start feature'} {shape_aspect_relationship.relying_shape_aspect is a part_template_definition whose description = 'trace template'}.

5.2.3.62 configuration_controlled_message_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The configuration_controlled_message_constraint rule constrains the value of the description of a descriptive_representation_item when it is used as an approval date, manufacturers code, part number, or revision identifier.

EXPRESS specification:

```
* )
( *
RULE configuration_controlled_message_constraint FOR
( descriptive_representation_item );
END_RULE; -- configuration_controlled_message_constraint
* )
( *
```

Informal propositions:

IP1: IF descriptive_representation_item\representation_item.name = 'approval date' the descriptive_representation_item shall be an item in a representation that is referenced in an approval_assignment and the value of SELF.description shall be the concatenation of the value of date.year_component + ',' + the value of calendar_date.month_component + ',' + the value of calendar_date.day_component + ',' + the value of local_time.hour_component + ',' + the value of local_time.minute_component + ',' + the value of local_time.second_component + ',' + the value of coordinated_universal_time_offset.hour_offset + ',' + the value of coordinated_universal_time_offset.minute_offset + ',' + the value of coordinated_universal_time_offset.sense.

IP2: IF descriptive_representation_item\representative_item.name = 'manufacturers code' the descriptive_representation_item shall be an item in a representation that is referenced in an organization_assignment and the value of SELF.description shall be identical to that of organization.id.

IP3: IF descriptive_representative_item\representative_item.name = 'part number' the descriptive_representation_item shall be an item in a representation that is referenced in a property_definition_representation which has a definition.definition.formation that is a product_definition_formation and the value of SELF.description shall be identical to that of product_definition_formation.of_product.id.

IP4: IF descriptive_representative_item\representative_item.name = 'revision identifier' the descriptive_representation_item shall be an item in a representation which is referenced in a property_definition_representation that has a definition.definition.formation that is a product_definition_formation and the value of SELF.description shall be identical to that of product_definition_formation.id.

5.2.3.63 configuration_item_requires_person_organization

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The configuration_item_requires_person_organization rule assures that each configuration_item has an applied_person_and_organization_assignment or an applied_organization_assignment.

EXPRESS specification:

```
*)
RULE configuration_item_requires_person_organization FOR
  (configuration_item,
   applied_person_and_organization_assignment,
   applied_organization_assignment);
WHERE

  WR1: SIZEOF (QUERY (ci <* configuration_item |
    NOT ((SIZEOF (QUERY (apaoa <*
      applied_person_and_organization_assignment |
      ci IN apaoa.items )) = 1 ) OR
      (SIZEOF (QUERY (aoa <*
        applied_organization_assignment |
        ci IN aoa.items )) = 1 )))) = 0;
END_RULE; -- configuration_item_requires_person_organization
(*
```

Formal propositions:

WR1: For each instance of configuration_item, there shall be exactly one instance of applied_person_and_organization_assignment that contains the instance of configuration_item in its set of items.

5.2.3.64 conical_tolerance_zone_boundary_and_surface_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conical_tolerance_zone_boundary_and_surface_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a conical tolerance zone boundary and surface relationship.

EXPRESS specification:

```

*)
( *
RULE conical_tolerance_zone_boundary_and_surface_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- conical_tolerance_zone_boundary_and_surface_relationship -
          -- constraint
*)
( *

```

Informal propositions:

IP1: Each `shape_aspect_relationship` with a name of 'conical tolerance zone boundary and surface' shall have a `related_shape_aspect` that is either a `part_template_definition` or is on the boundary of the shape of the product.

IP2: Each `shape_aspect_relationship` with a name of 'conical tolerance zone boundary and surface' shall have a `relating_shape_aspect` that is a `tolerance_zone_boundary` with a description of 'conical'.

IP3: Each `shape_aspect_relationship` with a name of 'conical tolerance zone boundary and surface' shall have exactly one `property_definition` with a description of 'conical tolerance zone boundary size'.

5.2.3.65 `conical_tolerance_zone_boundary_definition_constraint`

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conical_tolerance_zone_boundary_definition_constraint` rule constrains the use of a `property_definition` when it is used as a conical tolerance zone boundary definition.

EXPRESS specification:

```

*)
( *
RULE conical_tolerance_zone_boundary_definition_constraint FOR
  (property_definition);
END_RULE; -- conical_tolerance_zone_boundary_definition_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall be a `property_definition` of a `tolerance_zone_boundary` that has a description of 'conical'.

IP2: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall relate to exactly two `property_definition` that have a description of 'conical tolerance zone boundary size' to specify its 'boundary size'.

IP3: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall relate to exactly two `property_definition` that have a description of 'conical tolerance zone boundary size' to specify its 'boundary size' and each of the `property_definition` that have a description of 'conical tolerance zone boundary size' shall be related as a 'defined boundary' to exactly one `physical_unit_geometric_tolerance`, and each of these `physical_unit_geometric_tolerance` shall have the same name.

IP4: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall relate to exactly two `property_definition` that have a description of 'conical tolerance zone boundary size' to specify its 'boundary size' and each of the `property_definition` that have a description of 'conical tolerance zone boundary size' shall be related as a 'defined boundary' to exactly one `physical_unit_geometric_tolerance`, and each of these `physical_unit_geometric_tolerance` shall relate to the same `property_definition` with a description of either 'datum system property with material conditions' or 'datum system property without material conditions' to specify their 'referenced datum system definition'.

IP5: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall relate to exactly two `property_definition` that have a description of 'conical tolerance zone boundary size' to specify its 'boundary size' and each of the `property_definition` that have a description of 'conical tolerance zone boundary size' shall be related as a 'defined boundary' to exactly one `physical_unit_geometric_tolerance`, and each of these `physical_unit_geometric_tolerance` either shall not be modified_`geometric_tolerance` or shall have the same modifier.

IP6: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall relate to exactly two `property_definition` that have a description of 'conical tolerance zone boundary size' to specify its 'boundary size' and each of the `property_definition` that have a description of 'conical tolerance zone boundary size' shall be related as a 'defined boundary' to exactly one `physical_unit_geometric_tolerance`, and each of these `physical_unit_geometric_tolerance` shall specify the same instance of `shape_aspect` as their `toleranced_shape_aspect`, and that `shape_aspect` shall either be a `centre_of_symmetry` with a description of 'axis', or a `composite_shape_aspect` in which all the 'composing' `shape_aspects` are of type `centre_of_symmetry` and have a description of 'axis'.

IP7: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary' shall be a `property_definition` of a `tolerance_zone_boundary` that has a description of 'conical' and that `tolerance_zone_boundary` shall be specified as the first of the boundaries of a `tolerance_zone_definition` that specifies as its zone a `tolerance_zone` that has a description of 'conical'.

5.2.3.66 conical_tolerance_zone_boundary_diametrical_size_characteristic_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conical_tolerance_zone_boundary_diametrical_size_characteristic_constraint` rule constrains the use of a `property_definition` when it is used as a conical tolerance zone boundary diametrical size characteristic.

EXPRESS specification:

```

*)
( *
RULE
conical_tolerance_zone_boundary_diametrical_size_characteristic_constraint
FOR (property_definition);
END_RULE; -- conical_tolerance_zone_boundary_diametrical_size_-
          -- characteristic_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary size' shall be a `property_definition` of a `shape_aspect_relationship` that has a name of 'conical tolerance zone boundary and surface'.

IP2: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary size' shall be related as a 'boundary size' to exactly one `property_definition` that has a description of 'conical tolerance zone boundary'.

IP3: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary size' shall have a representation that contains exactly one `length_measure_with_unit` that has a name of 'conical tolerance zone boundary value', and that instance of `length_measure_with_unit` shall be the same instance of `length_measure_with_unit` as specified as the magnitude by the `physical_unit_geometric_tolerance` that relates to the instance of `property_definition` that has a description of 'conical tolerance zone boundary size' to specify its 'defined boundary'.

IP4: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary size' shall be related as a 'defined boundary' to exactly one `physical_unit_geometric_tolerance`, and that `physical_unit_geometric_tolerance` shall be either a `position_tolerance`, a `parallelism_tolerance`, a `perpendicularity_tolerance`, or an `angularity_tolerance`.

IP5: Each instance of `property_definition` that has a description of 'conical tolerance zone boundary size' shall be related as a 'boundary size' to exactly one `property_definition` that has a description of 'conical tolerance zone boundary' and that `property_definition` shall be applied to the same instance of `tolerance_zone_boundary` as specified as the `relating_shape_aspect` by the `shape_aspect_relationship` that has a name of 'conical tolerance zone boundary and surface' and to which the instance of `property_definition` is applied.

5.2.3.67 conical_tolerance_zone_definition_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `conical_tolerance_zone_definition_constraint` rule constrains the use of a `tolerance_zone_definition` when it is used as a conical tolerance zone definition.

EXPRESS specification:

```
* )
( *
RULE conical_tolerance_zone_definition_constraint FOR
  (tolerance_zone_definition);
END_RULE; -- conical_tolerance_zone_definition_constraint
* )
( *
```

Informal propositions:

IP1: Each `tolerance_zone_definition` that specifies a `tolerance_zone` with a description of 'conical' shall have exactly one boundary that is a `tolerance_zone_boundary` with a description of 'conical' and that `tolerance_zone_boundary` shall have exactly one `property_definition` that has a description of 'conical tolerance zone boundary'.

IP2: Each `tolerance_zone_definition` that specifies a `tolerance_zone` with a description of 'conical' as its zone, shall have that `tolerance_zone` specify exactly two `physical_unit_geometric_tolerance`, as its defining tolerance, and these `physical_unit_geometric_tolerance` shall be the same `physical_unit_geometric_tolerance` that relate to the `property_definitions` with a description of 'conical tolerance zone boundary size' that are applied to `shape_aspect_relationships` with names of 'conical tolerance zone boundary' and 'surface' that specify the `tolerance_zone_boundary`, as the `relating_shape_aspect`, that is specified as the boundaries of the `tolerance_zone_definition`.

5.2.3.68 connection_zone_constraint

The `connection_zone_constraint` rule constrains the use of `shape_aspect` and `shape_representation` when implementing the ARM concepts of `Connection_zone` and the associated colour.

EXPRESS specification:

```
* )
RULE connection_zone_constraint FOR ( shape_aspect, representation );
WHERE
```

```

WR1: sizeof( query( sa <* shape_aspect |
    (sa\shape_aspect.description = 'connection zone') AND NOT
    (sizeof (usedin (sa, '')) >= 1))) = 0;

WR2: sizeof( query( sa <* shape_aspect |
    (sa\shape_aspect.description = 'connection zone') AND
    NOT(sizeof( query( sar <* usedin(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (sar\shape_aspect_relationship.name IN [
    'connection',
    'minimum attachment region size',
    'maximum attachment region size',
    'external connection area',
    'external connection zone',
    'internal connection zone',
    'mounting area',
    'terminal connection zone'
    ]))) >= 1))) = 0;
END_RULE; -- connection_zone_constraint
( *

```

Formal propositions:

WR1: Each instance of shape_aspect with a description of 'connection zone' shall be referenced at least once.

WR2: Each instance of shape_aspect with a description of 'connection zone' shall be referenced by at least one shape_aspect_relationship with a name of 'connection', 'minimum attachment region size', 'maximum attachment region size', 'external connection area', 'external connection zone', 'internal connection zone', 'mounting area', or 'terminal connection zone'.

Informal propositions:

IP1: Each instance of shape_aspect with a description of 'connection zone' that is represented by a shape_representation that is capable of representing topological information shall be represented by a shape_representation that describes the connection zone as being a set of connected points.

EXAMPLE 1 If a manifold_surface_shape_representation were to represent a connection zone, there would only be one shell in the representation.

5.2.3.69 contract_unique_constraint

The contract_unique_constraint rule constrains the contract.name population members to satisfy the uniqueness requirement of the ARM application object Contract.

EXPRESS specification:

```
*)  
  
RULE contract_unique_constraint FOR (contract);  
  
LOCAL  
  r : BAG OF contract := QUERY (r <* contract | TRUE);  
  pass : BOOLEAN := TRUE;  
  name_bag : BAG OF string := [];  
END_LOCAL;  
  
REPEAT i := 1 to SIZEOF (r) by 1;  
  IF EXISTS( r[i]\contract.name ) THEN  
    IF (r[i]\contract.name IN name_bag) THEN  
      pass := FALSE;  
      ESCAPE;  
    ELSE  
      name_bag := name_bag + r[i]\contract.name;  
    END_IF;  
  END_IF;  
END_REPEAT;  
  
WHERE  
  WR1 : pass;  
  
END_RULE; -- contract_unique_constraint  
  
(*
```

Formal propositions:

WR1: The name shall be unique within a population of contract.

5.2.3.70 curve_extension_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The curve_extension_constraint rule limits the values that the extension for curve_style_with_extension shall be.

EXPRESS specification:

```

*)
( *
RULE curve_extension_constraint for ( curve_style_with_extension );
END_RULE; -- curve_extension_constraint
*)
( *

```

Informal propositions:

IP1: The allowed values for extension are: $+ 0.5 \cdot \text{width} \pm \text{delta}$, $0 \pm \text{delta}$, $- 0.5 \cdot \text{width} \pm \text{delta}$.
 where delta is uncertainty_measure_with_unit, assigned by user.

5.2.3.71 curve_style_font_and_scaling_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The curve_style_font_and_scaling_unique_constraint rule constrains the curve_style_font_and_scaling name attribute in a population to satisfy the uniqueness requirement of the ARM application object Curve_style_font_and_scaling.

EXPRESS specification:

*)

```

RULE curve_style_font_and_scaling_unique_constraint
  FOR (curve_style_font_and_scaling);

LOCAL
  r : BAG OF curve_style_font_and_scaling :=
    QUERY (r <* curve_style_font_and_scaling | TRUE);
  pass : BOOLEAN := TRUE;
  name_bag : BAG OF string := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF (r) by 1;
  IF EXISTS( r[i]\curve_style_font_and_scaling.name ) THEN
    IF (r[i]\curve_style_font_and_scaling.name IN name_bag) THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      name_bag := name_bag + r[i]\curve_style_font_and_scaling.name;
    END_IF;
  END_IF;
END_REPEAT;

WHERE
WR1 : pass;

END_RULE; -- curve_style_font_and_scaling_unique_constraint

( *
```

Formal propositions:

WR1: The name shall be unique within a population of curve_style_font_and_scaling.

5.2.3.72 curve_style_requires_length_measure_with_unit

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The curve_style_requires_length_measure_with_unit rule constrains the entity pointed to by the curve_width attribute of a curve_style.

EXPRESS specification:

```

*)
RULE curve_style_requires_length_measure_with_unit FOR ( curve_style );
WHERE

    WR1: SIZEOF(QUERY(cs <* curve_style |
        NOT('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(cs))
        )) = 0;
END_RULE; -- curve_style_requires_length_measure_with_unit
( *

```

Formal propositions:

WR1: A curve_style.curve_width shall be a length_measure_with_unit.

5.2.3.73 curve_style_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The curve_style_unique_constraint rule constrains the representation population members, that fill the role of ARM curve_style, so that the ARM name shall be unique within the population.

EXPRESS specification:

```

*)
RULE curve_style_unique_constraint FOR
  ( representation );
LOCAL
  cs : BAG OF representation := QUERY( r <* representation |
  ((r\representation.name = 'curve style parameters') AND (r.context_of_items\representation.
  'curve style parametric context')) );
  pass : BOOLEAN := TRUE;
  desc_bag : BAG OF STRING := [];
END_LOCAL;

REPEAT i := 1 to SIZEOF(cs) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  REPEAT j := 1 to SIZEOF(cs[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(cs[i].items[j])) AND
    (cs[i].items[j]\representation_item.name = 'curve style name') ) THEN
      IF EXISTS( cs[i].items[j]\descriptive_representation_item.description ) THEN
        IF ( cs[i].items[j]\descriptive_representation_item.description IN desc_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          desc_bag := desc_bag + cs[i].items[j]\descriptive_representation_item.description;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- curve_style_unique_constraint
(

```

Formal propositions:

WR1: Every instance of representation with a name of 'curve style parameters' must have a unique descriptive_representation_item.description attribute where the descriptive_representation_item has a name of 'curve style name' and is a member of the representation's set of items.

5.2.3.74 date_and_time_require_minute_and_second

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `date_and_time_require_minute_and_second` rule constrains `date_and_time` to include a minute and second component.

EXPRESS specification:

```
*)
RULE date_and_time_require_minute_and_second FOR ( date_and_time );
WHERE

    WR1: SIZEOF(QUERY(dat <* date_and_time |
                      NOT(Exists(dat.time_component.minute_component)))) = 0;

    WR2: SIZEOF(QUERY(dat <* date_and_time |
                      NOT(Exists(dat.time_component.second_component)))) = 0;
END_RULE; -- date_and_time_require_minute_and_second
(*
```

Formal propositions:

WR1: A `date_and_time` shall have a `time_component.minute_component`.

WR2: A `date_and_time` shall have a `time_component.second_component`.

5.2.3.75 **date_constraint**

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `date_constraint` rule specifies that all instances of `date` shall have a `year_number` that has 4 digits.

EXPRESS specification:

```
*)
RULE date_constraint FOR (date);
WHERE

    WR1: SIZEOF (QUERY (d <* date |
                       d.year_component < 1856)) = 0;
END_RULE; -- date_constraint
(*
```

Formal propositions:

WR1: For each instance of `date`, the `year_component` shall be greater than or equal to 1856.

5.2.3.76 datum_axis_and_tolerance_zone_orienting_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_axis_and_tolerance_zone_orienting_relationship_constraint rule constrains the use of a shape_aspect_relationship when it is used as a datum axis and tolerance zone orienting relationship.

EXPRESS specification:

```
*)
(*
RULE datum_axis_and_tolerance_zone_orienting_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_axis_and_tolerance_zone_orienting_relationship_constraint
*)
(*
```

Informal propositions:

IP1: Each shape_aspect_relationship with a name of 'datum axis and tolerance zone' shall have a related_shape_aspect that is a physical_unit_datum with a description of 'axis'.

IP2: Each shape_aspect_relationship with a name of 'datum axis and tolerance zone' shall have a relating_shape_aspect that is a tolerance_zone.

IP3: Each shape_aspect_relationship with a name of 'datum axis and tolerance zone' shall have exactly one property_definition with a description of 'datum axis related orientation'.

5.2.3.77 datum_axis_related_orientation_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_axis_related_orientation_constraint rule constrains the use of a property_definition when it is used as a datum axis related orientation.

EXPRESS specification:

```
*)
(*
RULE datum_axis_related_orientation_constraint FOR
  (property_definition);
END_RULE; -- datum_axis_related_orientation_constraint
*)
(*
```

Informal propositions:

IP1: Each instance of property_definition that has a description of 'datum axis related orientation' shall be a property_definition of a shape_aspect_relationship that has a name of 'datum axis and tolerance zone'.

IP2: Each instance of property_definition that has a description of 'datum axis related orientation' shall have a name of either 'centred on arc' or 'centred on radial'.

IP3: Each instance of property_definition that has a description of 'datum axis related orientation' shall be related as a 'zone orientation' to exactly one property_definition that has a description of 'tolerance zone orientation'.

5.2.3.78 datum_based_direction_vector_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_based_direction_vector_constraint rule constrains the use of a property_definition when it is used as a datum based direction vector.

EXPRESS specification:

```
*)
(*
RULE datum_based_direction_vector_constraint FOR
  (property_definition);
END_RULE; -- datum_based_direction_vector_constraint
*)
(*
```

Informal propositions:

IP1: Each instance of property_definition that has a description of 'datum direction property' shall be a property_definition of a physical_unit_datum that has a description of 'plane'.

IP2: Each instance of property_definition that has a description of 'datum direction property' shall have a representation that contains exactly one axis1_placement that has a name of 'orientation'.

IP3: Each instance of property_definition that has a description of 'datum direction property' shall have a name of either 'positive direction vector' or 'negative direction vector'.

IP4: Each instance of property_definition that has a description of 'datum direction property' shall be related as a 'element direction vector' to at least one property_definition that has a description of 'direction element'.

5.2.3.79 datum_based_length_measure_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_based_length_measure_constraint rule constrain the usage of a representation and the items of a representation when it is used as maximum component height or clearance.

EXPRESS specification:

```
*)
(*
RULE datum_based_length_measure_constraint FOR ( representation );
END_RULE; -- datum_based_length_measure_constraint
*)
(*
```

Informal propositions:

IP1: Each representation with a name = ('maximum positive component height') ('maximum negative component height') ('maximum mounting clearance') ('minimum mounting clearance') shall have one items that is a length_measure_with_unit.name = 'measure'.

IP2: Each representation with a name = ('maximum positive component height') ('maximum negative component height') ('maximum mounting clearance') ('minimum mounting clearance') shall have one items that is a direction.name = 'measure orientation' or that is a descriptive_representation_item.name = 'measure orientation'.

IP3: Each representation with a name = ('maximum positive component height') ('maximum negative component height') ('maximum mounting clearance') ('minimum mounting clearance') shall be the used_representation of one property_definition_representation that has a definition.definition that is a datum.

5.2.3.80 datum_based_vector_orientation_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_based_vector_orientation_constraint rule constrains the use of a property_definition when it is used as a datum based vector orientation.

EXPRESS specification:

```

*)
( *
RULE datum_based_vector_orientation_constraint FOR
  (property_definition);
END_RULE; -- datum_based_vector_orientation_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of property_definition that has a description of 'datum based vector orientation' shall be a property_definition of a shape_aspect that has a description of 'vector'.

IP2: Each instance of property_definition that has a description of 'datum based vector orientation' shall relate to exactly two property_definition that have a description of 'direction element' to specify its 'direction component'.

IP3: Each instance of property_definition that has a description of 'datum based vector orientation' either shall be related as a 'zone orientation' to at least one property_definition that has a description of 'tolerance zone orientation', shall be related as a 'measurement orientation' to at least one property_definition that has a description of 'dimensional location property' and that is a property_definition of a dimensional_location_with_direction, or shall be related as a 'measurement orientation' to at least one property_definition that has a description of 'dimensional location property' and that is a property_definition of an angular_dimension_with_orientation.

IP4: Each instance of property_definition that has a description of 'datum based vector orientation' shall relate to exactly two property_definition that have a description of 'direction element' to specify its 'direction component' and each of these shall be a property_definition of instances of shape_aspect_relationship that have a name of 'datum plane and half plane relationship' and each of these shape_aspect_relationship shall specify a different physical_unit_datum that has a description of 'plane' as their related_shape_aspect.

IP5: Each instance of property_definition that has a description of 'datum based vector orientation' shall relate to exactly two property_definition that have a description of 'direction element' to specify its 'direction component' and each of these shall be a property_definition of instances of shape_aspect_relationship that have a name of 'datum plane and half plane relationship' and each of these shape_aspect_relationship shall specify a different physical_unit_datum that has a description of 'plane' as their related_shape_aspect and these physical_unit_datum shall be related to the same datum_reference_frame as 'reference plane'.

IP6: Each instance of `property_definition` that has a description of 'direction element' shall relate to exactly one `property_definition` that has a description of 'datum direction property' to specify its 'half datum plane direction vector', and this `property_definition` shall be applied to a `physical_unit_datum` that has a description of 'plane'.

Additionally, each instance of `property_definition` that has a description of 'direction element' shall relate to exactly one `property_definition` that has a description of 'datum direction property' to specify its 'element direction vector', and this `property_definition` shall be applied to a `physical_unit_datum` that has a description of 'plane'.

Finally, each instance of `property_definition` that has a description of 'direction element' shall be a `property_definition` of a `shape_aspect_relationship` that has a name of 'datum plane and half plane relationship' and that `shape_aspect_relationship` shall have a `related_shape_aspect` that is of type `physical_unit_datum` and has a description of 'plane'.

None of these `physical_unit_datum` shall be the same.

IP7: Each instance of `property_definition` that has a description of 'direction element' shall relate to exactly one `property_definition` that has a description of 'datum direction property' to specify its 'half datum plane direction vector', this `property_definition` shall be a `property_definition` of a `physical_unit_datum` that has a description of 'plane'.

Additionally, each instance of `property_definition` that has a description of 'direction element' shall relate to exactly one `property_definition` that has a description of 'datum direction property' to specify its 'element direction vector', this `property_definition` shall be a `property_definition` of a `physical_unit_datum` that has a description of 'plane'.

Finally, each instance of `property_definition` that has a description of 'direction element' shall be a `property_definition` of a `shape_aspect_relationship` that has a name of 'datum plane and half plane relationship' and that `shape_aspect_relationship` shall have a `related_shape_aspect` of type `physical_unit_datum` that has a description of 'plane'.

These `physical_unit_datum` shall be related to the same `datum_reference_frame` as 'reference plane'.

5.2.3.81 datum_feature_material_condition_property_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_material_condition_property_constraint` rule constrains the use of a `property_definition` when it is used as a datum feature material condition property.

EXPRESS specification:

```

*)
( *
RULE datum_feature_material_condition_property_constraint FOR
  (property_definition);
END_RULE; -- datum_feature_material_condition_property_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'datum feature material condition property' shall be a `property_definition` of a `shape_aspect_relationship` that has a name of 'datum feature usage in datum system'.

IP2: Each instance of `property_definition` that has a description of 'datum feature material condition property' shall be related as a 'applied material condition property' to exactly one `property_definition` that has a description of 'datum system property with material conditions'.

IP3: Each instance of `property_definition` that has a description of 'datum feature material condition property' shall have a name of either 'least material requirement' or 'maximum material principle'.

IP4: Each instance of `property_definition` that has a description of 'datum feature material condition property' shall be related as an 'applied material condition property' to exactly one `property_definition` that has a description of 'datum system property with material conditions' and that `property_definition` shall be applied to the same instance of `datum_system` as specified as the `relating_shape_aspect` by the `shape_aspect_relationship` that has a name of 'datum feature usage in datum system' and to which the instance of `property_definition` is applied.

5.2.3.82 datum_feature_usage_in_common_datum_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_usage_in_common_datum_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a datum feature usage in common datum relationship.

EXPRESS specification:

```

*)
( *
RULE datum_feature_usage_in_common_datum_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_feature_usage_in_common_datum_relationship_constraint
*)
( *

```

Informal propositions:

IP1: Each `shape_aspect_relationship` with a name of 'datum feature usage in common datum' shall have a relating `shape_aspect` that is a `physical_unit_datum` that has a name of 'common datum'.

5.2.3.83 datum_feature_usage_in_datum_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_usage_in_datum_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a datum feature usage in datum relationship.

EXPRESS specification:

```

*)
( *
RULE datum_feature_usage_in_datum_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_feature_usage_in_datum_relationship_constraint
*)
( *
```

Informal propositions:

IP1: Each `shape_aspect_relationship` with a name of either 'datum feature usage in common datum' or 'datum feature usage in single datum' shall have a related `shape_aspect` that is a `physical_unit_datum_feature`.

IP2: Each `shape_aspect_relationship` with a name of either 'datum feature usage in common datum' or 'datum feature usage in single datum' shall have a relating `shape_aspect` that is a `physical_unit_datum`.

IP3: Each `shape_aspect_relationship` with a name of either 'datum feature usage in common datum' or 'datum feature usage in single datum' shall specify a unique combination of relating `shape_aspect` and related `shape_aspect`.

5.2.3.84 datum_feature_usage_in_datum_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_usage_in_datum_relationship_unique_constraint` rule constrains `shape_aspect_relationship` population members, that fill the role of ARM `datum_feature_usage_in_datum_relationship`, so that the combination of ARM relating `shape_element` and ARM related `shape_element` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE datum_feature_usage_in_datum_relationship_unique_constraint FOR
  ( shape_aspect_relationship );
LOCAL
  dfuidr : BAG OF shape_aspect_relationship :=
    QUERY( sar <* shape_aspect_relationship |
      ((sar\shape_aspect_relationship.name = 'datum feature usage in common datum') OR
      (sar\shape_aspect_relationship.name = 'datum feature usage in single datum')) );
  pudf_bag : BAG OF physical_unit_datum_feature := [];
  sar_bag : BAG OF shape_aspect_relationship;
  pass : BOOLEAN := TRUE;
  pud_bag : BAG OF physical_unit_datum;
END_LOCAL;

REPEAT i := 1 to SIZEOF(dfuidr) by 1;
  IF EXISTS( dfuidr[i].relating_shape_aspect ) THEN
    IF( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM' IN TYPEOF(dfuidr[i].relating_shape_aspect) ) THEN
      IF ( NOT ( dfuidr[i].relating_shape_aspect IN pud_bag ) ) THEN
        pud_bag := pud_bag + dfuidr[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pud_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  pudf_bag := [];
  sar_bag :=
    QUERY( sar <* dfuidr | ((sar.relating_shape_aspect :=: pud_bag[i]) AND
    ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM_FEATURE' IN TYPEOF(sar.related_shape_aspect))) );
  REPEAT j := 1 to SIZEOF(sar_bag) by 1;
    IF EXISTS( sar_bag[j].related_shape_aspect ) THEN
      IF ( sar_bag[j].related_shape_aspect IN pudf_bag ) THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        pudf_bag := pudf_bag + sar_bag[j].related_shape_aspect;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- datum_feature_usage_in_datum_relationship_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of `shape_aspect_relationship` with a description of either 'datum feature usage in common datum' or 'datum feature usage in single datum' must have a unique combination of `relating_shape_aspect` and `related_shape_aspect` attributes.

5.2.3.85 datum_feature_usage_in_datum_system_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_usage_in_datum_system_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a datum feature usage in datum system relationship.

EXPRESS specification:

```

*)
( *
RULE datum_feature_usage_in_datum_system_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_feature_usage_in_datum_system_relationship_constraint
*)
( *
```

Informal propositions:

IP1: Each `shape_aspect_relationship` with a name of 'datum feature usage in datum system' shall have a `related_shape_aspect` that is a `physical_unit_datum_feature`.

IP2: Each `shape_aspect_relationship` with a name of 'datum feature usage in datum system' shall have a `relating_shape_aspect` that is a `datum_system`.

IP3: Each `shape_aspect_relationship` with a name of 'datum feature usage in datum system' shall have exactly one `property_definition` with a description of 'datum feature material condition property'.

IP4: Each `shape_aspect_relationship` with a name of 'datum feature usage in datum system' shall have a `related_shape_aspect` that is a `physical_unit_datum_feature` that is in the set of `physical_unit_datum_feature` that are related as either 'datum feature usage in common datum' or 'datum feature usage in single datum' to the `physical_unit_datum` that are related as the 'datum usage in datum system' to the `datum_system` that is the `relating_shape_aspect`.

5.2.3.86 datum_feature_usage_in_single_datum_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_feature_usage_in_single_datum_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a datum feature usage in single datum relationship.

EXPRESS specification:

```

*)
( *
RULE datum_feature_usage_in_single_datum_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_feature_usage_in_single_datum_relationship_constraint
*)
( *

```

Informal propositions:

IP1: Each shape_aspect_relationship with a name of 'datum feature usage in single datum' shall have a relating_shape_aspect that is a physical_unit_datum that has a name of 'single datum'.

5.2.3.87 datum_precedence_assignment_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_precedence_assignment_constraint rule constrains the use of a property_definition when it is used as a datum precedence assignment.

EXPRESS specification:

```

*)
( *
RULE datum_precedence_assignment_constraint FOR
  (property_definition);
END_RULE; -- datum_precedence_assignment_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of property_definition that has a description of 'datum precedence assignment' shall be a property_definition of a shape_aspect_relationship that has a name of 'datum usage in datum system'.

IP2: Each instance of property_definition that has a description of 'datum precedence assignment' shall be related as a 'assigned datum precedence' to exactly one property_definition that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions'.

IP3: Each instance of `property_definition` that has a description of 'datum precedence assignment' shall have a name of either 'primary', 'secondary', or 'tertiary'.

IP4: Each instance of `property_definition` that has a description of 'datum precedence assignment' shall be related as an 'assigned datum precedence' to exactly one `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' and that `property_definition` shall be applied to the same instance of `datum_system` as specified as the `relating_shape_aspect` by the `shape_aspect_relationship` that has a name of 'datum usage in datum system' and to which the instance of `property_definition` is applied.

5.2.3.88 datum_precedence_assignment_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_precedence_assignment_unique_constraint` rule constrains `property_definition` population members, that fill the role of ARM `datum_precedence_assignment`, so that the combination of ARM name and ARM `datum_system_conditions_definition` shall be unique within the population.

EXPRESS specification:

```

*)
RULE datum_precedence_assignment_unique_constraint FOR
  ( property_definition );
LOCAL
  dpa : BAG OF property_definition := QUERY( pd <* property_definition |
(pd\property_definition.description = 'datum precedence assignment') );
  name_bag : BAG OF STRING := [];
  pd_bag : BAG OF property_definition;
  pdr_bag : BAG OF property_definition_relationship;
  pass : BOOLEAN := TRUE;
  pds : BAG OF property_definition;
END_LOCAL;

REPEAT i := 1 to SIZEOF(dpa) by 1;
  IF EXISTS( dpa[i]\property_definition.name ) THEN
    IF( NOT( dpa[i]\property_definition.name IN name_bag ) ) THEN
      name_bag := name_bag + dpa[i]\property_definition.name;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(name_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  pds := [];
  pd_bag := QUERY( pd <* dpa | (pd\property_definition.name = name_bag[i]) );
  REPEAT j := 1 to SIZEOF(pd_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    pdr_bag := QUERY( pdr <* USEDIN(pd_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION' ) |
((pdr\property_definition_relationship.name = 'assigned datum precedence') AND
(pdr.relating_property_definition\property_definition.description IN [
'datum system property with material conditions',
'datum system property without material conditions']))) );
    REPEAT k := 1 to SIZEOF(pdr_bag) by 1;
      IF EXISTS( pdr_bag[k].relating_property_definition ) THEN
        IF ( pdr_bag[k].relating_property_definition IN pds ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pds := pds + pdr_bag[k].relating_property_definition;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;
WHERE

```

```

    WR1: pass;
END_RULE; -- datum_precedence_assignment_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of `property_definition` with a description of 'datum precedence assignment' must have a unique combination of name attribute and `property_definition` entity with a description of either 'datum system property with material conditions' or 'datum system property without material conditions' where the `property_definitions` are related by a `property_definition_relationship` with a name of 'assigned datum precedence'.

5.2.3.89 datum_system_defining_conditions_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_system_defining_conditions_constraint` rule constrains the use of a `property_definition` when it is used as a datum system defining conditions.

EXPRESS specification:

```

*)
( *
RULE datum_system_defining_conditions_constraint FOR
    (property_definition);
END_RULE; -- datum_system_defining_conditions_constraint
*)
( *

```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' shall be a `property_definition` of a `datum_system`.

IP2: Each instance of `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' shall relate to exactly one, two, or three `property_definition` that have a description of 'datum precedence assignment' to specify its 'assigned datum precedence'.

IP3: Each instance of `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' that relates, as 'assigned datum precedence', to a `property_definition` that has a description of 'datum precedence assignment' and has a name of 'tertiary' shall also relate, as 'assigned datum precedence', to a `property_definition` that has a description of 'datum precedence assignment' and has a name of 'secondary'.

IP4: Each instance of `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' that relates, as 'assigned datum precedence', to a `property_definition` that has a description of 'datum precedence assignment' and has a name of 'secondary' shall also relate, as 'assigned datum precedence', to a `property_definition` that has a description of 'datum precedence assignment' and has a name of 'primary'.

IP5: Each instance of `property_definition` that has a description of 'datum system property with material conditions' shall be related as a 'referenced datum system definition' to at least one `geometric_tolerance_with_specified_datum_system`.

IP6: Each instance of `property_definition` that has a description of 'datum system property without material conditions' either shall be related as a 'referenced datum system definition' to at least one `geometric_tolerance_with_specified_datum_system` or shall be related as a 'referenced datum system property' to at least one `property_definition` that has a description of 'dimensional location property' and that is a `property_definition` for a `datum_system_based_dimensional_location`.

IP7: Each instance of `property_definition` that has a description of either 'datum system property with material conditions' or 'datum system property without material conditions' shall relate to exactly one, two, or three `property_definition` that have a description of 'datum precedence assignment' to specify its 'assigned datum precedence' and those `property_definition` shall have unique names.

5.2.3.90 datum_system_defining_conditions_with_material_conditions_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_system_defining_conditions_with_material_conditions_constraint` rule constrains the use of a `property_definition` when it is used as a datum system defining conditions with material conditions.

EXPRESS specification:

```
* )
( *
RULE datum_system_defining_conditions_with_material_conditions_constraint
FOR (property_definition);
END_RULE; -- datum_system_defining_conditions_with_material_-
          -- conditions_constraint
* )
( *
```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'datum system property with material conditions' shall relate to at least one `property_definition` that has a description of 'datum feature material condition property' to specify its 'applied material condition property'.

5.2.3.91 datum_system_defining_conditions_without_material_conditions_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_system_defining_conditions_without_material_conditions_constraint` rule constrains the use of a `property_definition` when it is used as a datum system defining conditions without material conditions.

EXPRESS specification:

```
* )
( *
RULE
datum_system_defining_conditions_without_material_conditions_constraint FOR
    (property_definition);
END_RULE; -- datum_system_defining_conditions_without_material_
           -- conditions_constraint
* )
( *
```

Informal propositions:

IP1: Each instance of `property_definition` that has a description of 'datum system property without material conditions' shall relate to exactly zero `property_definition` that have a description of 'datum feature material condition property' to specify its 'applied material condition property'.

5.2.3.92 datum_target_usage_in_datum_target_set_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_target_usage_in_datum_target_set_relationship_constraint` rule constrains the use of a `shape_aspect_relationship` when it is used as a datum target usage in datum target set relationship.

EXPRESS specification:

```

*)
( *
RULE datum_target_usage_in_datum_target_set_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_target_usage_in_datum_target_set_relationship_constraint
*)
( *

```

Informal propositions:

IP1: Each shape_aspect_relationship with a name of 'datum target usage' shall have a related_shape_aspect that is a physical_unit_datum_target.

IP2: Each shape_aspect_relationship with a name of 'datum target usage' shall have a relating_shape_aspect that is a physical_unit_datum_target_set.

IP3: Each shape_aspect_relationship with a name of 'datum target usage' shall have a description that is expressed only by digits and represents a positive integer.

IP4: Each shape_aspect_relationship with a name of 'datum target usage' shall specify a unique combination of relating_shape_aspect and related_shape_aspect.

IP5: Each shape_aspect_relationship with a name of 'datum target usage' shall specify a unique combination of relating_shape_aspect and description.

5.2.3.93 datum_target_usage_in_datum_target_set_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_target_usage_in_datum_target_set_relationship_unique_constraint rule constrains shape_aspect_relationship population members, that fill the role of ARM datum_target_usage_in_datum_target_set_relationship, so that the combination of ARM used_datum_target and ARM defined_datum_target attributes and that the combination of ARM datum_target_number and ARM defined_datum_target attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE datum_target_usage_in_datum_target_set_relationship_unique_constraint
FOR ( shape_aspect_relationship );
LOCAL
    dtuids : BAG OF shape_aspect_relationship :=
        QUERY( sar <* shape_aspect_relationship |
(sar\shape_aspect_relationship.name = 'datum target usage'));
    pudts_bag : BAG OF physical_unit_datum_target_set := [];
    sar_bag : BAG OF shape_aspect_relationship;
    pass1 : BOOLEAN := TRUE;
    pudt_bag : BAG OF physical_unit_datum_target;
    pass2 : BOOLEAN := TRUE;
    num_bag : BAG OF STRING;
END_LOCAL;

REPEAT i := 1 to SIZEOF(dtuids) by 1;
    IF EXISTS( dtuids[i].relating_shape_aspect ) THEN
        IF( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_TARGET_SET' IN
        TYPEOF(dtuids[i].relating_shape_aspect) ) THEN
            IF ( NOT ( dtuids[i].relating_shape_aspect IN pudts_bag ) ) THEN
                pudts_bag := pudts_bag + dtuids[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pudts_bag) by 1;
    IF ( NOT (pass1 AND pass2) ) THEN ESCAPE;
    END_IF;
    num_bag := [];
    pudt_bag := [];
    sar_bag :=
        QUERY( sar <* dtuids | (sar.relating_shape_aspect :=: pudts_bag[i]) );
    REPEAT j := 1 to SIZEOF(sar_bag) by 1;
        IF ( NOT (pass1 AND pass2) ) THEN ESCAPE;
        END_IF;
        IF ( pass1 AND
            ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_TARGET' IN
            TYPEOF(sar_bag[j].related_shape_aspect)) ) THEN
            IF EXISTS( sar_bag[j].related_shape_aspect ) THEN
                IF ( sar_bag[j].related_shape_aspect IN pudt_bag ) THEN
                    pass1 := FALSE;
                ELSE
                    pudt_bag := pudt_bag + sar_bag[j].related_shape_aspect;
                END_IF;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;

```

```

    IF ( pass2 ) THEN
      IF EXISTS( sar_bag[j]\shape_aspect_relationship.description ) THEN
        IF ( sar_bag[j]\shape_aspect_relationship.description IN num_bag ) THEN
          pass2 := FALSE;
        ELSE
          num_bag := num_bag + sar_bag[j]\shape_aspect_relationship.description;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass1;

  WR2: pass2;
END_RULE; -- datum_target_usage_in_datum_target_set_relationship_unique_-
          -- constraint
( *
```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'datum target usage' must have a unique combination of related_shape_aspect and relating_shape_aspect attributes.

WR2: Every instance of shape_aspect_relationship with a name of 'datum target usage' must have a unique combination of description and relating_shape_aspect attributes.

5.2.3.94 datum_usage_in_datum_system_relationship_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The datum_usage_in_datum_system_relationship_constraint rule constrains the use of a shape_aspect_relationship when it is used as a datum usage in datum system relationship.

EXPRESS specification:

```

*)
( *
RULE datum_usage_in_datum_system_relationship_constraint FOR
  (shape_aspect_relationship);
END_RULE; -- datum_usage_in_datum_system_relationship_constraint
*)
( *
```

Informal propositions:

IP1: Each `shape_aspect_relationship` with a name of 'datum usage in datum system' shall have a `related_shape_aspect` that is a `physical_unit_datum`, and the name of `physical_unit_datum` shall be either 'common datum' or 'single datum'

IP2: Each `shape_aspect_relationship` with a name of 'datum usage in datum system' shall have a `relating_shape_aspect` that is a `datum_system`.

IP3: Each `shape_aspect_relationship` with a name of 'datum usage in datum system' shall have exactly one `property_definition` with a description of 'datum precedence assignment'.

IP4: Each `shape_aspect_relationship` with a name of 'datum usage in datum system' shall specify a unique combination of `relating_shape_aspect` and `related_shape_aspect`.

5.2.3.95 datum_usage_in_datum_system_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `datum_usage_in_datum_system_relationship_unique_constraint` rule constrains `shape_aspect_relationship` population members, that fill the role of ARM `datum_usage_in_datum_system_relationship`, so that the combination of ARM `established_datum_system` and ARM `used_datum` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE datum_usage_in_datum_system_relationship_unique_constraint FOR
  ( shape_aspect_relationship );
LOCAL
  duidsr : BAG OF shape_aspect_relationship :=
    QUERY( sar <* shape_aspect_relationship |
(sar\shape_aspect_relationship.name = 'datum usage in datum system'));
  ds_bag : BAG OF datum_system := [];
  sar_bag : BAG OF shape_aspect_relationship;
  pass : BOOLEAN := TRUE;
  pud_bag : BAG OF physical_unit_datum;
END_LOCAL;

REPEAT i := 1 to SIZEOF(duidsr) by 1;
  IF EXISTS( duidsr[i].relating_shape_aspect ) THEN
    IF( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_SYSTEM' IN TYPEOF(duidsr[i].relating_shape_aspect) ) THEN
      IF ( NOT ( duidsr[i].relating_shape_aspect IN ds_bag ) ) THEN
        ds_bag := ds_bag + duidsr[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

REPEAT i := 1 to SIZEOF(ds_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  pud_bag := [];
  sar_bag :=
    QUERY( sar <* duidsr | (sar.relating_shape_aspect := ds_bag[i]) );
  REPEAT j := 1 to SIZEOF(sar_bag) by 1;
    IF EXISTS( sar_bag[j].related_shape_aspect ) THEN
      IF ( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM' IN
        TYPEOF(sar_bag[j].related_shape_aspect) ) THEN
        IF ( sar_bag[j].related_shape_aspect IN pud_bag ) THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pud_bag := pud_bag + sar_bag[j].related_shape_aspect;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

```

```

    WR1: pass;
END_RULE; -- datum_usage_in_datum_system_relationship_unique_constraint
( *

```

Formal propositions:

WR1: Every instance of shape_aspect_relationship with a name of 'datum usage in datum system' must have a unique combination of related_shape_aspect and relating_shape_aspect attributes.

5.2.3.96 default_assembly_bond_shape_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The default_assembly_bond_shape_constraint rule constrains the usage of a shape_representation when it is used as a default assembly bond shape.

EXPRESS specification:

```

*)
RULE default_assembly_bond_shape_constraint FOR ( shape_representation );
WHERE

    WR1: SIZEOF(QUERY(sr <* shape_representation |
        (sr\representation.name = 'default assembly bond shape')
        AND NOT(SIZEOF(QUERY(pdr <* USEDIN(sr,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
            ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'SHAPE_ASPECT'
            IN TYPEOF(pdr.definition.definition)))) = 1))) = 0;
END_RULE; -- default_assembly_bond_shape_constraint
( *

```

Formal propositions:

WR1: Each shape_representation\representation.name = 'default assembly bond shape' shall be related to exactly one assembly_bond_definition.

5.2.3.97 default_passage_based_land_physical_template_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `default_passage_based_land_physical_template_unique_constraint` rule constrains `land_physical_template` population members, that fill the role of `ARM default_passage_based_land_physical_template`, so that the combination of `ARM of_passage_technology` and `ARM of_stratum_technology` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE default_passage_based_land_physical_template_unique_constraint FOR
  ( land_physical_template );
LOCAL
  dpblpt : BAG OF land_physical_template :=
QUERY( lpt <* land_physical_template |
(lpt\shape_aspect.description IN ['default via based',
'default attachment size and via based',
'default unsupported passage based',
'default component termination passage based',
'default attachment size and component termination passage based']) );
  sar_bag : BAG OF shape_aspect_relationship;
  pt_bag : BAG OF passage_technology := [];
  pdr_bag : BAG OF property_definition_relationship;
  pass : BOOLEAN := TRUE;
  st_bag : BAG OF stratum_technology;
END_LOCAL;

REPEAT i := 1 to SIZEOF(dpblpt) by 1;
  sar_bag := QUERY( sar <* USEDIN(dpblpt[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar\shape_aspect_relationship.name =
'technology usage') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY' IN TYPEOF(sar.related_shape_aspect)))) );
  REPEAT j := 1 to SIZEOF(sar_bag) by 1;
    IF EXISTS( sar_bag[j].related_shape_aspect ) THEN
      IF ( NOT ( sar_bag[j].related_shape_aspect IN pt_bag ) ) THEN
        pt_bag := pt_bag + sar_bag[j].related_shape_aspect;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pt_bag) by 1;
  IF ( NOT pass ) THEN ESCAPE;
  END_IF;
  st_bag := [];
  sar_bag := QUERY( sar <* USEDIN(pt_bag[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar\shape_aspect_relationship.name =
'technology usage') AND (sar.relatng_shape_aspect IN dpblpt)) );
  REPEAT j := 1 to SIZEOF(sar_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    pdr_bag :=
      QUERY( pdr <* USEDIN(sar_bag[j].relating_shape_aspect\shape_aspect.of_shape,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION') |

```

```

((pdr\property_definition_relationship.name = 'technology usage') AND
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY' IN
  TYPEOF(pdr.relatng_property_definition.definition))) );
REPEAT k := 1 to SIZEOF(pdr_bag) by 1;
  IF EXISTS( pdr_bag[k].relating_property_definition.definition ) THEN
    IF
      ( pdr_bag[k].relating_property_definition.definition IN st_bag )
      THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        st_bag := st_bag +
          pdr_bag[k].relating_property_definition.definition;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- default_passage_based_land_physical_template_unique_constraint
(*)

```

Formal propositions:

WR1: Every instance of `land_physical_template` with a description of either 'default via based', 'default attachment size and via based', 'default unsupported passage based', 'default component termination passage based' or 'default attachment size and component termination passage based' must have a unique combination of `passage_technology` and `stratum_technology` entities where the `passage_technology` is related to the `land_physical_template` by a `shape_aspect_relationship` with a name of 'technology usage' and where the `stratum_technology` is related to the `land_physical_template` by a `property_definition_relationship` with a name of 'technology usage'.

5.2.3.98 delete_design_object_management_relationship_unique_constraint

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `delete_design_object_management_relationship_unique_constraint` rule constrains the `delete_design_object_assignment` and `delete_design_object_request_assignment` population members, that fill the role of `ARM delete_design_object_management_relationship`, so that the combination of `ARM previous_design` and `ARM previous_design_object` attributes shall be unique within the population.

EXPRESS specification:

```

*)
RULE delete_design_object_management_relationship_unique_constraint FOR
  ( delete_design_object_assignment,
    delete_design_object_request_assignment );
LOCAL
  pdr_bag : BAG OF product_definition_relationship := [];
  pd_bag : BAG OF product_definition := [];
  ddoa_bag : BAG OF delete_design_object_assignment;
  ddora_bag : BAG OF delete_design_object_request_assignment;
  pass : BOOLEAN := TRUE;
  mdo_bag : BAG OF managed_design_object;
END_LOCAL;

REPEAT i := 1 to SIZEOF(delete_design_object_assignment) by 1;
  REPEAT j := 1 TO SIZEOF(delete_design_object_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP'
        IN TYPEOF(delete_design_object_assignment[i].items[j]))
      AND (delete_design_object_assignment[i].items[j].name =
        'design object deletion') ) THEN
      IF EXISTS( delete_design_object_assignment[i].items[j].
        relating_product_definition )
        THEN
          IF( NOT( delete_design_object_assignment[i].items[j].
            relating_product_definition
              IN pd_bag ) ) THEN
            pd_bag := pd_bag +
              delete_design_object_assignment[i].items[j].
                relating_product_definition;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

REPEAT i := 1 to SIZEOF(delete_design_object_request_assignment) by 1;
  REPEAT j := 1 TO
    SIZEOF(delete_design_object_request_assignment[i].items) by 1;
    IF ( ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP'
        IN TYPEOF(delete_design_object_request_assignment[i].items[j]))
      AND (delete_design_object_request_assignment[i].items[j].name =
        'design object deletion') ) THEN
      IF EXISTS
        ( delete_design_object_request_assignment[i].items[j].
          relating_product_definition )
        THEN
          IF( NOT
            ( delete_design_object_request_assignment[i].items[j].

```

```

relating_product_definition
    IN pd_bag ) ) THEN
    pd_bag := pd_bag +
        delete_design_object_request_assignment[i].items[j].
            relating_product_definition;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

REPEAT i := 1 to SIZEOF(pd_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    mdo_bag := [];
    pdr_bag := QUERY( pdr <* USEDIN(pd_bag[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        (pdr\product_definition_relationship.name = 'design object deletion
REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    ddoa_bag := QUERY( ddoa <* delete_design_object_assignment |
        (pdr_bag[j] IN ddoa.items) );
    REPEAT k := 1 to SIZEOF(ddoa_bag) by 1;
        IF ( NOT pass ) THEN ESCAPE;
        END_IF;
        REPEAT l := 1 to SIZEOF(ddoa_bag[k].items) by 1;
            IF EXISTS( ddoa_bag[k].items[l] ) THEN
                IF ( ddoa_bag[k].items[l] IN mdo_bag ) THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    mdo_bag := mdo_bag + ddoa_bag[k].items[l];
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;
REPEAT j := 1 to SIZEOF(pdr_bag) by 1;
    IF ( NOT pass ) THEN ESCAPE;
    END_IF;
    ddora_bag :=
        QUERY( ddora <* delete_design_object_request_assignment |
            (pdr_bag[j] IN ddora.items) );
    REPEAT k := 1 to SIZEOF(ddora_bag) by 1;
        IF ( NOT pass ) THEN ESCAPE;
        END_IF;
        REPEAT l := 1 to SIZEOF(ddora_bag[k].items) by 1;
            IF EXISTS( ddora_bag[k].items[l] ) THEN
                IF ( ddora_bag[k].items[l] IN mdo_bag ) THEN
                    pass := FALSE;

```

```

        ESCAPE;
      ELSE
        mdo_bag := mdo_bag + ddora_bag[k].items[l];
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE

  WR1: pass;
END_RULE; -- delete_design_object_management_relationship_unique_constraint
(*)

```

Formal propositions:

WR1: Every instance of `delete_design_object_assignment` and `delete_design_object_request_assignment` that has a common `product_definition` that is the `relating_product_definition` of a `product_definition_relationship` that has a name of 'design object deletion' where the `product_definition_relationship` is a member of the `items` attribute of the `delete_design_object_assignment` and `delete_design_object_request_assignment` must have unique managed_design object entities in their `items` attributes.

5.2.3.99 **dependent_instantiable_action_directive**

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The `dependent_instantiable_action_directive` rule specifies that all instances of `action_directive` are dependent on the usage to define another entity.

EXPRESS specification:

```

*)
RULE dependent_instantiable_action_directive FOR (action_directive);
WHERE

  WR1: SIZEOF (QUERY (ad <* action_directive |
    NOT (SIZEOF (USEDIN (ad, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_action_directive
(*)

```

Formal propositions:

WR1: For each instance of `action_directive`, there shall be a reference to the `action_directive` instance from an attribute of another entity.

5.2.3.100 dependent_instantiable_approval_status

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_approval_status rule specifies that all instances of approval_status are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_status FOR (approval_status);
WHERE

WR1: SIZEOF (QUERY (ast <* approval_status |
    NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_approval_status
(*
```

Formal propositions:

WR1: For each instance of approval_status, there shall be a reference to the approval_status instance from an attribute of another entity.

5.2.3.101 dependent_instantiable_certification_type

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_certification_type rule specifies that all instances of certification_type are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_certification_type FOR (certification_type);
WHERE

WR1: SIZEOF (QUERY (ct <* certification_type |
    NOT (SIZEOF (USEDIN (ct, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_certification_type
(*
```

Formal propositions:

WR1: For each instance of certification_type, there shall be a reference to the certification_type instance from an attribute of another entity.

5.2.3.102 dependent_instantiable_contract_type

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_contract_type rule specifies that all instances of contract_type are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_contract_type FOR (contract_type);
WHERE

    WR1: sizeof(QUERY ( ct <* contract_type |
        NOT (sizeof (USEDIN (ct, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_contract_type
(*
```

Formal propositions:

WR1: For each instance of contract_type, there shall be a reference to the contract_type instance from an attribute of another entity.

5.2.3.103 dependent_instantiable_date

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_date rule specifies that all instances of date are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date FOR (date);
WHERE
```

```

WR1: SIZEOF (QUERY (dt <* date |
    NOT (SIZEOF(USEDIN (dt, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_date
( *

```

Formal propositions:

WR1: For each instance of date, there shall be a reference to the date instance from an attribute of another entity.

5.2.3.104 dependent_instantiable_date_role

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_date_role rule specifies that all instances of date_role are dependent on the usage to define another entity.

EXPRESS specification:

```

*)
RULE dependent_instantiable_date_role FOR (date_role);
WHERE

```

```

WR1: SIZEOF (QUERY (dr <* date_role |
    NOT (SIZEOF (USEDIN (dr, '')) >= 1))) = 0;
END_RULE; -- dependent_instantiable_date_role
( *

```

Formal propositions:

WR1: For each instance of date_role, there shall be a reference to the date_role instance from an attribute of another entity.

5.2.3.105 dependent_instantiable_date_time_role

NOTE This rule may not have been updated to agree with the changes in clause 4.2.

The dependent_instantiable_date_time_role rule specifies that all instances of date_time_role are dependent on the usage to define another entity.

EXPRESS specification:

```
* )  
RULE dependent_instantiable_date_time_role FOR (date_time_role);  
WHERE  
  
    WR1: SIZEOF (QUERY (dtr <* date_time_role |  
        NOT (SIZEOF (USEDIN (dtr, '')) >= 1))) = 0;  
END_RULE; -- dependent_instantiable_date_time_role  
(*
```

Formal propositions:

WR1: For each instance of date_time_role, there shall be a reference to the date_time_role instance from an attribute of another entity.

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods:

- ISO 10303-21;
- ISO 10303-22.

Requirements with respect to implementation methods are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in annex D.

This Part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes:

- Class 1 - Device Functional and Physical Characterization;
- Class 2 - Interconnect Technology Constraints;
- Class 3 - Assembly Technology Constraints;
- Class 4 - Assembly Functional Requirements;
- Class 5 - Assembly Physical Requirements;
- Class 6 - Interconnect Functional Requirements;
- Class 7 - Interconnect Physical Requirements;
- Class 8 - Assembly Physical Design;
- Class 9 - Interconnect Design;
- Class 10 - Interconnect Design (Microwave);
- Class 11 - Geometric Dimensioning and Tolerancing;
- Class 12 - Product Rule;
- Class 13 - Functional Decomposition;

- Class 14 - Package Functional and Physical Characterization;
- Class 15 - Geometrically Bounded Surface Model;
- Class 16 - Wireframe Model with Topology;
- Class 17 - Advanced Boundary Representation;
- Class 18 - Constructive Solid Geometry;
- Class 19 - Extruded Solid;
- Class 20 - Geometrically Bounded 2d Wireframe Model;
- Class 21 - Wireframe 2d Model with Topology;
- Class 22 - Curve 2d;
- Class 23 - Basic Curve 2d;
- Class 24 - Laminate Assembly Design ;
- Class 25 - Connection Zone Based Model Extraction;
- Class 26 - Functional Specification;
- Class 27 - Physical Unit Physical Characterization;
- Class 28 - Packaged Part White Box Model;
- Class 29 - Printed Part Functional and Physical Characterization;
- Class 30 - Open Shell Model ;
- Class 31 - Assembly Functional Interface Requirements;
- Class 32 - Assembly Physical Interface Requirements;
- Class 33 - Assembly Requirement Allocation;
- Class 34 - Interconnect Requirement Allocation;
- Class 35 - Functional Requirement Allocation;
- Class 36 - Product Requirement Allocation;

- Class 37 - Datum Difference Based Model Definition ;
- Class 38 - Datum Difference Based Characteristic;
- Class 39 - Design Management .

Conformance to this part of ISO 10303 requires conformance to one of the following:

- any combination of the conformance classes 1 through 7;
- conformance class 8;
- any combination of the conformance classes 9, 10, 24;
- conformance class 12 ;
- any combination of the conformance classes 13, 14;
- any combination of the conformance classes 24 through 29 ;
- any combination of the conformance classes 31 through 39 .

Conformance classes 15 through 23, and class 30, are shape representation conformance classes for this part of ISO 10303, and shall be used with other conformance classes of this part of ISO 10303. Conformance to a particular conformance class requires that all AIM entities, types, and associated constraints defined as part of that class be supported. Conformance to a particular conformance class requires conformance to each conformance class included in that class. All entities specified, either directly or indirectly, by required attributes of the required AIM entities shall be supported. Conformance to a particular conformance class requires that all ARM constraints for UoFs implemented by this class be supported.

Support for a particular conformance class requires support of all the options specified in this class.

NOTE The following information is provided as a list rather than table format.

6.1 Device Functional and Physical Characterization

Device Data includes following information: device black box model, package data, functional data, environmental constraints, performance data, simulation models. Formal encapsulation of external data type definitions is made for parametric data and analysis models. Pin mapping is provided to ensure consistency between device views. Shape, material, technology, tolerance on shape and on parameters, units are included. Shapes included may be categorized according to discipline (e.g. thermal analysis, vibration analysis), and analysis environment (e.g., design, assembly, end-user). Geometric Dimensioning and Tolerancing is included. The functional data includes two distinct data sets: hierarchical functional decomposition into behavioural elements embedded in a network represented at each level by a nodal formulation; collection of device terminals (e.g., connector terminals, printed circuit board interface areas, jumper ends) that implement the functionality of a network node. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- altered_part;
- analytical_model;
- analytical_representation;
- assembly_module_usage_view;
- authorization;
- bare_die;
- characteristic;
- datum_based_characteristic;
- device_marking;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_with_nodal_representation;
- functional_definition_to_physical_unit_usage_view_mapping;
- functional_specification;

- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- library_item;
- material_product;
- package;
- packaged_part_black_box_model;
- part_feature;
- part_feature_to_analytical_model_port_mapping;
- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- product_identification;
- shape;
- signal;
- source_control;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/device_functional_and_physical_characterization_class.exp..

6.2 Interconnect Technology Constraints

This Data includes all information provided to the design team by fabrication vendors from which may be derived default land and passage definitions, based on the desired yield for fabrication and assembly processes. Typical Data includes minimum annular ring, maximum passage aspect ratio, minimum deposition thickness, maximum terminal size supported for through hole technology class, and other critical material processing properties. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_with_nodal_representation;
- functional_definition_to_printed_part_mapping;
- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- interconnect_marking;
- library_item;
- manufacturing_technology;
- material_product;
- part_template;

- part_template_shape;
- planned_characteristic;
- printed_part_template;
- product_identification;
- requirement_decomposition;
- shape;
- signal;
- source_control;
- stratum;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/interconnect_technology_constraints_class.exp..

6.3 Assembly Technology Constraints

This Data includes all information provided to the design team by assembly vendors from which may be derived constraints on which packages may be selected, mounting arrangements to be specified, permitted mounting areas, clearances, etc. Typical bond shape for each unique assembly process is available. Extensive use of formal encapsulation of external data type definitions is made for parametric data. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- assembly_marking;
- assembly_physical_interface_requirement;

- assembly_technology;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- library_item;
- physical_unit_shape;
- planned_characteristic;
- product_identification;
- requirement_decomposition;
- shape;
- source_control.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/assembly_technology_constraints_class.exp..

6.4 Assembly Functional Requirements

This Data includes all information required to specify the behaviour of the assembly, including interface definition. Explicit structural definition is provided for the functional network, including representations of usage view and design view, representations of folded and elaborated network hierarchy. This includes gate allocation information. Explicit allocation of each functional network node to the implementing component is included. Extensive use of formal encapsulation is provided for signal definition, mathematical models, etc. This data includes information required to support embedded components in an interconnect product. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- assembly_functional_interface_requirement;
- assembly_functional_requirement_allocation;
- assembly_module_usage_view;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_to_assembly_design;
- functional_decomposition_with_nodal_representation;
- functional_decomposition_with_nodal_representation_to_assembly_mapping;
- functional_definition_to_physical_unit_usage_view_mapping;
- functional_design_elaboration;
- functional_specification;
- functional_unit_analytical_representation;

- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- library_item;
- packaged_part_black_box_model;
- planned_characteristic;
- product_identification;
- requirement_decomposition;
- signal;
- source_control.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- applied_group_assignment;
- approval;
- area_measure_with_unit;
- assembly_definition;
- assembly_group_component_definition;

- assembly_joint;
- assembly_module_interface_terminal;
- assembly_module_join_terminal;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- bus_element_link;
- bus_structural_definition;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- certification;
- change_from_design_object_assignment;
- change_from_design_object_request_assignment;
- change_to_design_object_assignment;
- characterized_product_category;
- colour_rgb;
- component_definition;

- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- composite_signal_property_relationship;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;

- design_make_from_relationship;
- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electrical_network;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_assembly_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;

- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- group_shape_aspect;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- interface_component_definition;
- interface_mounted_join;
- laminate_component_interface_terminal;

- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;
- library_defined_assembly_definition;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_connectivity_relationship;
- make_from_functional_unit_terminal_definition_relationship;
- make_from_model_port_relationship;
- managed_design_object;
- mapped_item;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;

- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- network_node_definition;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_connector_terminal_relationship;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_connected_terminals_definition;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;

- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_node_requirement_to_implementing_component_allocation;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_part_template_link;
- printed_part_template_terminal;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;

- product_definition_with_associated_documents;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- requirements_property_group;
- resistance_measure_with_unit;
- rule_definition;
- scalar_terminal_definition_link;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;

- shape_aspect;
- shape_aspect_relationship;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- standard_uncertainty;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- thermal_feature;
- thermal_network;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.5 Assembly Physical Requirements

This Data includes all information provided to the design team that may be represented by shape data, including customer requirements, and technology selected to implement those requirements. Explicit allowed volumetric shape, external connection locations are included. Component grouping, keepout, keepin, etc is included. This data includes information required to support embedded components in an interconnect product. Configuration management information and design management information is provided. Preferred parts and packages may be specified by inclusion in design library. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- assembly_bill_of_material;
- assembly_component_placement_requirements;
- assembly_module_usage_view;
- assembly_physical_interface_requirement;
- assembly_physical_requirement_allocation;
- authorization;
- characteristic;
- datum_based_characteristic;
- document;
- end_item_identification;
- external_definition;
- product_identification;
- requirement_decomposition;
- shape;
- source_control;

- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/assembly_physical_requirements_class.exp..

6.6 Interconnect Functional Requirements

This Data includes the device data from conformance class 1 (only the functional device information) specific to an interconnect product (e.g., pcb, substrate, flex board). The functional view of Devices which are fabricated as part of the interconnect product fabrication process (e.g., printed inductors, printed connectors, printed capacitors) are included in the functional definition. Devices which are embedded are considered to be external to the interconnect product since they are not fabricated as part of the product and their shape does not directly contribute to the shape definition of the interconnect product. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_to_interconnect_design;
- functional_decomposition_with_nodal_representation;
- functional_decomposition_with_nodal_representation_to_interconnect_mapping;
- functional_definition_to_physical_unit_usage_view_mapping;
- functional_definition_to_printed_part_mapping;
- functional_design_elaboration;

- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- interconnect_module_usage_view;
- library_item;
- planned_characteristic;
- product_identification;
- requirement_decomposition;
- signal;
- source_control;
- test_probe.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- applied_group_assignment;
- approval;

- area_measure_with_unit;
- assembly_group_component_definition;
- assembly_module_terminal;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- bus_element_link;
- bus_structural_definition;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- certification;
- change_from_design_object_assignment;
- change_from_design_object_request_assignment;
- change_to_design_object_assignment;
- characterized_product_category;
- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;

- component_functional_unit;
- component_interface_terminal;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- composite_signal_property_relationship;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- curve;
- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- design_make_from_relationship;

- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- edge_segment_vertex;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electrical_network;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_interconnect_definition;
- externally_defined_package;
- externally_defined_physical_unit;
- externally_defined_product_definition;

- fabrication_joint;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- interconnect_definition;
- interconnect_module_component_surface_feature;
- interconnect_module_cutout_segment_surface;
- interconnect_module_edge_segment_surface;
- interconnect_module_interface_terminal;
- interconnect_module_join_terminal;
- interconnect_module_stratum_based_terminal;

- interconnect_module_terminal;
- laminate_component_interface_terminal;
- land;
- land_physical_template;
- land_template_terminal;
- layer;
- layer_connection_point;
- length_measure_with_unit;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_interconnect_definition;
- library_defined_package;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_connectivity_relationship;
- make_from_functional_unit_terminal_definition_relationship;
- make_from_model_port_relationship;
- managed_design_object;
- mass_measure_with_unit;
- material_designation;

- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- network_node_definition;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;

- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- plated_inter_stratum_feature;
- plated_passage;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_component;
- printed_component_link;
- printed_part_template_link;
- printed_part_template_material;
- printed_part_template_material_link;
- printed_part_template_terminal;
- probe_access_area;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;

- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;
- product_definition_with_associated_documents;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- requirements_property_group;
- resistance_measure_with_unit;
- routed_printed_component;
- rule_definition;
- scalar_terminal_definition_link;

- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- standard_uncertainty;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- template_material_cross_section_boundary;
- thermal_feature;
- thermal_network;
- thermodynamic_temperature_measure_with_unit;

- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.7 Interconnect Physical Requirements

This Data includes all information related to shape and position requirements. Trace, via and other passage spacing, keepin, keepout, etc. is included. Explicit allowed volumetric shape, required connection locations, material specifications, etc. are included. Those layout items whose placement is driven by thermal considerations or electromagnetic interference may be specified. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- datum_based_characteristic;
- document;
- end_item_identification;
- external_definition;
- interconnect_bill_of_material;
- interconnect_interface_component;
- interconnect_module_usage_view;
- interconnect_placement_requirements;
- product_identification;

- requirement_decomposition;
- shape;
- source_control;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/interconnect_physical_requirements_class.exp..

6.8 Assembly Physical Design

This data includes all data that defines the physical relationships between the components in the assembly. This data includes all the components that exist in the assembly, and specifies those that provide physical interfaces to the next level of assembly. Several types of assembly joint may be specified. Design re-use is explicitly supported with traceability. Complete traceability back to requirements is provided. This data identifies those components that do not meet design requirements. Appropriate elements of Geometric Dimensioning and Tolerancing are provided. This data includes both design view and usage view of the assembly. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- assembly_bill_of_material;
- assembly_component;
- assembly_component_grouping;
- assembly_component_placement_requirements;
- assembly_design_information_assignment_to_usage_view;
- assembly_interface_component;
- assembly_marking;

- assembly_module_design;
- assembly_module_usage_view;
- assembly_physical_interface_requirement;
- assembly_physical_requirement_allocation;
- assembly_shape;
- authorization;
- characteristic;
- datum_based_characteristic;
- document;
- end_item_identification;
- external_definition;
- library_item;
- material_product;
- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- planned_characteristic;
- planned_effectivity;
- product_identification;
- requirement_decomposition;
- shape;
- source_control;
- styled_curve;

- styled_text.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- alternate_product_relationship;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- annotation_curve_occurrence;
- annotation_fill_area;
- annotation_fill_area_occurrence;
- applied_promissory_usage_in_product_model_assignment;
- approval;
- area_measure_with_unit;
- array_placement_group_component_definition;
- assembly_bond_definition;
- assembly_component_usage;
- assembly_definition;
- assembly_group_component_definition;
- assembly_group_component_definition_placement_link;

- assembly_joint;
- assembly_module_interface_terminal;
- assembly_module_join_terminal;
- assembly_module_macro_component_join_terminal;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- boolean_result;
- cable_component;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- cartesian_transformation_operator_2d;
- celsius_temperature_measure_with_unit;
- certification;
- characterized_product_category;
- circular_area;
- colour_rgb;
- component_definition;

- component_feature;
- component_feature_relationship;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_location;
- component_mating_constraint_condition;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_curve;
- composite_curve_segment;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_effectivity;
- configuration_item;
- connection_zone_based_assembly_joint;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- csg_2d_shape_representation;
- curve;

- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- data_set_representation_item;
- date_and_time;
- dated_effectivity;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- direction;
- discrete_shield;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- elliptic_area;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;

- expanded_uncertainty;
- external_definition;
- externally_defined_assembly_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_hatch_style;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- externally_defined_text_font;
- externally_defined_tile;
- externally_defined_tile_style;
- fill_area_style;
- fill_area_style_colour;
- fill_area_style_hatching;
- fill_area_style_tile_coloured_region;
- fill_area_style_tile_curve_with_style;
- fill_area_style_tile_symbol_with_style;
- fill_area_style_tiles;
- footprint_definition;

- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- group_product_definition;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- interconnect_module_component_surface_feature;
- interconnect_module_join_terminal;
- interconnect_module_macro_component_join_terminal;
- interconnect_module_terminal;
- interface_component_definition;
- interface_mounted_join;
- interfaced_group_component_definition;
- laminate_component_interface_terminal;
- laminate_group_component_make_from_relationship;

- layer_connection_point;
- length_measure_with_unit;
- library_defined_assembly_definition;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;
- library_defined_physical_unit;
- linear_array_component_definition_link;
- linear_array_placement_group_component_definition;
- lot_effectivity;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_feature_relationship;
- make_from_model_port_relationship;
- make_from_usage_option;
- managed_design_object;
- manifold_subsurface_shape_representation;
- manifold_surface_shape_representation;
- mapped_item;

- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- mounting_restriction_area;
- mounting_restriction_volume;
- next_assembly_usage_occurrence;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;

- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- polygonal_area;
- power_measure_with_unit;
- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship;
- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship;
- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- primitive_2d;
- product;
- product_category_relationship;

- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;
- product_definition_with_associated_documents;
- product_material_composition_relationship;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- promissory_usage_occurrence;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- quantified_assembly_component_usage;
- ratio_measure_with_unit;
- rectangular_area;
- rectangular_array_placement_group_component_definition;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;

- representation_relationship_with_transformation;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- requirements_property_group;
- resistance_measure_with_unit;
- routed_physical_component;
- rule_definition;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- serial_numbered_effectivity;
- shape_aspect;
- shape_aspect_relationship;
- shape_representation;
- shape_representation_relationship;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- specified_higher_usage_occurrence;
- standard_uncertainty;
- stratum;

- stratum_feature;
- stratum_technology;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- text_literal;
- thermal_component;
- thermal_feature;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- trimmed_curve;
- type_qualifier;
- uncertainty_qualifier;
- vector;
- volume_measure_with_unit;
- volume_shape_intersection.

6.9 Interconnect Design

This data provides functional and physical layout information sufficient to allow manufacture and test of an interconnect. Design re-use is explicitly supported with traceability. Complete traceability back to requirements is provided. Product connection requirements, shape requirements, product specifications, process specifications, material specifications including manufacturing view of stackup, Geometric Dimensioning and Tolerancing are provided. This data identifies those elements that do not meet design requirements. This data includes both design view and usage view of the interconnect product. Support for specification of signal prioritization is provided. Support for specifying the explicit network topology to be implemented is provided. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- datum_based_characteristic;
- document;
- end_item_identification;
- external_definition;
- fabrication_joint;
- form_features;
- inter_stratum_feature_template;
- interconnect_bill_of_material;
- interconnect_component_grouping;
- interconnect_design_information_assignment_to_usage_view;
- interconnect_interface_component;
- interconnect_marking;

- interconnect_module_connection_routing;
- interconnect_module_design;
- interconnect_module_edge;
- interconnect_module_usage_view;
- interconnect_placement_requirements;
- interconnect_shape;
- land;
- library_item;
- material_product;
- part_template;
- part_template_shape;
- physical_network;
- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- planned_characteristic;
- planned_effectivity;
- printed_part_template;
- product_identification;
- requirement_decomposition;
- shape;
- source_control;
- stratum;

- stratum_feature;
- stratum_position;
- styled_curve;
- styled_text;
- test_probe.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- alternate_product_relationship;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- annotation_curve_occurrence;
- annotation_fill_area;
- annotation_fill_area_occurrence;
- applied_promissory_usage_in_product_model_assignment;
- approval;
- area_measure_with_unit;
- array_placement_group_component_definition;
- array_placement_group_component_shape_aspect;

- assembly_bond_definition;
- assembly_component_usage;
- assembly_group_component_definition;
- assembly_group_component_definition_placement_link;
- assembly_joint;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- boolean_result;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- cartesian_transformation_operator_2d;
- celsius_temperature_measure_with_unit;
- certification;
- characterized_product_category;
- circular_area;
- colour_rgb;
- component_definition;

- component_feature;
- component_feature_relationship;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_location;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_curve;
- composite_curve_segment;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- conductive_interconnect_element_terminal_link;
- configuration_effectivity;
- configuration_item;
- connected_area_component;
- connection_zone_based_fabrication_joint;
- connection_zone_interface_plane_relationship;
- connection_zone_map_identification;
- contract;
- coordinated_representation_item;

- csg_2d_shape_representation;
- curve;
- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- cutout_edge_segment;
- data_set_representation_item;
- date_and_time;
- dated_effectivity;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- design_layer_type_specific_padstack_definition;
- direction;
- discrete_shield;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- edge_segment_cross_section;
- edge_segment_vertex;
- ee_specification;
- electric_charge_measure_with_unit;

- electric_current_measure_with_unit;
- electrical_isolation_component_shape_aspect;
- electrical_isolation_removal_template_definition;
- electromotive_force_measure_with_unit;
- elliptic_area;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_hatch_style;
- externally_defined_interconnect_definition;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- externally_defined_text_font;
- externally_defined_tile;
- externally_defined_tile_style;

- fabrication_joint;
- fiducial;
- fiducial_stratum_feature;
- fill_area_style;
- fill_area_style_colour;
- fill_area_style_hatching;
- fill_area_style_tile_coloured_region;
- fill_area_style_tile_curve_with_style;
- fill_area_style_tile_symbol_with_style;
- fill_area_style_tiles;
- filled_area_material_removal_component_shape_aspect;
- footprint_definition;
- footprint_occurrence;
- footprint_occurrence_shape_aspect_relationship;
- force_measure_with_unit;
- frequency_measure_with_unit;
- frozen_assigned_item;
- frozen_assignment;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;

- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- group_product_definition;
- group_shape_aspect;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- inter_stratum_feature_edge_segment_template;
- inter_stratum_feature_edge_template;
- interconnect_definition;
- interconnect_module_component_surface_feature;
- interconnect_module_cutout_segment_surface;
- interconnect_module_edge_segment;
- interconnect_module_edge_segment_surface;
- interconnect_module_interface_terminal;
- interconnect_module_join_terminal;
- interconnect_module_macro_component_join_terminal;
- interconnect_module_stratum_based_terminal;
- interconnect_module_terminal;
- interface_access_component_shape_aspect;
- interface_access_material_removal_component_shape_aspect;

- interface_component_definition;
- interfaced_group_component_definition;
- item_defined_transformation;
- join_shape_aspect;
- laminate_component_interface_terminal;
- laminate_group_component_make_from_relationship;
- land;
- land_physical_template;
- land_template_terminal;
- layer;
- layer_connection_point;
- layer_connection_point_link;
- layout_macro_component;
- layout_macro_definition;
- layout_macro_definition_terminal_to_usage_terminal_assignment;
- layout_macro_floor_plan_template;
- length_measure_with_unit;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_interconnect_definition;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;

- library_defined_physical_unit;
- linear_array_component_definition_link;
- linear_array_placement_group_component_definition;
- lot_effectivity;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_connectivity_relationship;
- make_from_feature_relationship;
- make_from_model_port_relationship;
- make_from_usage_option;
- managed_design_object;
- manifold_constraining_context_dependent_shape_representation;
- manifold_subsurface_shape_representation;
- manifold_surface_shape_representation;
- mapped_item;
- mass_measure_with_unit;
- material_designation;
- material_property;
- material_removal_component_shape_aspect;
- measure_qualification;
- measure_representation_item;

- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- mounting_restriction_area;
- mounting_restriction_volume;
- multi_layer_component_shape_aspect;
- multi_layer_material_removal_component_shape_aspect;
- multi_layer_stratum_feature;
- next_assembly_usage_occurrence;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- padstack_occurrence;
- padstack_occurrence_shape_aspect_relationship;
- parameter_assignment_representation;
- part_connected_terminals_definition;

- part_mounting_feature;
- part_template_definition;
- part_text_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_network;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- plated_cutout_edge_segment;
- plated_inter_stratum_feature;
- plated_interconnect_module_edge_segment;
- plated_passage;
- polygonal_area;
- power_measure_with_unit;
- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship;
- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship;

- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- primitive_2d;
- printed_component;
- printed_component_link;
- printed_connector_template_terminal_relationship;
- printed_part_cross_section_template_terminal;
- printed_part_template_connected_terminals_definition;
- printed_part_template_link;
- printed_part_template_material;
- printed_part_template_material_link;
- printed_part_template_terminal;
- probe_access_area;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;

- product_definition_with_associated_documents;
- product_material_composition_relationship;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- promissory_usage_occurrence;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- quantified_assembly_component_usage;
- ratio_measure_with_unit;
- rectangular_area;
- rectangular_array_placement_group_component_definition;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- representation_relationship_with_transformation;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- requirements_property_group;
- resistance_measure_with_unit;

- routed_printed_component;
- rule_definition;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- sequential_laminate_passage_based_fabrication_joint;
- serial_numbered_effectivity;
- shape_aspect;
- shape_aspect_relationship;
- shape_dimension_representation;
- shape_modification;
- shape_representation;
- shape_representation_relationship;
- shape_representation_relationship_with_transformation;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- standard_uncertainty;
- stratum;
- stratum_concept_relationship;
- stratum_feature;
- stratum_occurrence_specific_padstack_definition;

- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- stratum_type_independent_padstack_definition;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- template_material_cross_section_boundary;
- template_stratum_assignment;
- text_literal;
- thermal_component;
- thermal_feature;
- thermal_isolation_component_shape_aspect;
- thermal_isolation_removal_template_definition;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- trimmed_curve;
- type_qualifier;
- uncertainty_qualifier;
- usage_view_connection_zone_terminal_shape_relationship;
- vector;
- volume_measure_with_unit;
- volume_shape_intersection.

6.10 Interconnect Design (Microwave)

This data is similar to Class 9, with the exception that the metallization may be considered to be microstrip or stripline, with a specified shape element of the cross-section(i.e., point, edge, cutting plane) acting as the terminal(pair, port) of the line or component. Use of formal external definitions is provided to link in models with the product definition data. Analytical model terminals may be typed as having distributed parameters. This data identifies those elements that do not meet design requirements. This data includes both design view and usage view of the interconnect product. Support for specification of signal prioritization is provided. Support for specifying the explicit network topology to be implemented is provided. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- datum_based_characteristic;
- document;
- end_item_identification;
- external_definition;
- fabrication_joint;
- form_features;
- inter_stratum_feature_template;
- interconnect_bill_of_material;
- interconnect_component_grouping;
- interconnect_design_information_assignment_to_usage_view;
- interconnect_interface_component;
- interconnect_marking;

- interconnect_module_connection_routing;
- interconnect_module_design;
- interconnect_module_edge;
- interconnect_module_usage_view;
- interconnect_placement_requirements;
- interconnect_shape;
- land;
- library_item;
- material_product;
- part_template;
- part_template_shape;
- physical_network;
- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- planned_characteristic;
- planned_effectivity;
- printed_part_template;
- product_identification;
- requirement_decomposition;
- shape;
- source_control;
- stratum;

- stratum_feature;
- stratum_position;
- styled_curve;
- styled_text;
- test_probe.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- alternate_product_relationship;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- annotation_curve_occurrence;
- annotation_fill_area;
- annotation_fill_area_occurrence;
- applied_promissory_usage_in_product_model_assignment;
- approval;
- area_measure_with_unit;
- array_placement_group_component_definition;
- array_placement_group_component_shape_aspect;

- assembly_bond_definition;
- assembly_component_usage;
- assembly_group_component_definition;
- assembly_group_component_definition_placement_link;
- assembly_joint;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- boolean_result;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- cartesian_transformation_operator_2d;
- celsius_temperature_measure_with_unit;
- certification;
- characterized_product_category;
- circular_area;
- colour_rgb;
- component_definition;

- component_feature;
- component_feature_relationship;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_location;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_curve;
- composite_curve_segment;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- conductive_interconnect_element_terminal_link;
- configuration_effectivity;
- configuration_item;
- connected_area_component;
- connection_zone_based_fabrication_joint;
- connection_zone_interface_plane_relationship;
- connection_zone_map_identification;
- contract;
- coordinated_representation_item;

- csg_2d_shape_representation;
- curve;
- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- cutout_edge_segment;
- data_set_representation_item;
- date_and_time;
- dated_effectivity;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- design_layer_type_specific_padstack_definition;
- direction;
- discrete_shield;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- edge_segment_cross_section;
- edge_segment_vertex;
- ee_specification;
- electric_charge_measure_with_unit;

- electric_current_measure_with_unit;
- electrical_isolation_component_shape_aspect;
- electrical_isolation_removal_template_definition;
- electromotive_force_measure_with_unit;
- elliptic_area;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_hatch_style;
- externally_defined_interconnect_definition;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- externally_defined_text_font;
- externally_defined_tile;
- externally_defined_tile_style;

- fabrication_joint;
- fiducial;
- fiducial_stratum_feature;
- fill_area_style;
- fill_area_style_colour;
- fill_area_style_hatching;
- fill_area_style_tile_coloured_region;
- fill_area_style_tile_curve_with_style;
- fill_area_style_tile_symbol_with_style;
- fill_area_style_tiles;
- filled_area_material_removal_component_shape_aspect;
- footprint_definition;
- footprint_occurrence;
- footprint_occurrence_shape_aspect_relationship;
- force_measure_with_unit;
- frequency_measure_with_unit;
- frozen_assigned_item;
- frozen_assignment;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;

- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- group_product_definition;
- group_shape_aspect;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- inter_stratum_feature_edge_segment_template;
- inter_stratum_feature_edge_template;
- interconnect_definition;
- interconnect_module_component_surface_feature;
- interconnect_module_cutout_segment_surface;
- interconnect_module_edge_segment;
- interconnect_module_edge_segment_surface;
- interconnect_module_interface_terminal;
- interconnect_module_join_terminal;
- interconnect_module_macro_component_join_terminal;
- interconnect_module_stratum_based_terminal;
- interconnect_module_terminal;
- interface_access_component_shape_aspect;
- interface_access_material_removal_component_shape_aspect;

- interface_component_definition;
- interfaced_group_component_definition;
- item_defined_transformation;
- join_shape_aspect;
- laminate_component_interface_terminal;
- laminate_group_component_make_from_relationship;
- land;
- land_physical_template;
- land_template_terminal;
- layer;
- layer_connection_point;
- layer_connection_point_link;
- layout_macro_component;
- layout_macro_definition;
- layout_macro_definition_terminal_to_usage_terminal_assignment;
- layout_macro_floor_plan_template;
- length_measure_with_unit;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_interconnect_definition;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;

- library_defined_physical_unit;
- linear_array_component_definition_link;
- linear_array_placement_group_component_definition;
- lot_effectivity;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_connectivity_relationship;
- make_from_feature_relationship;
- make_from_model_port_relationship;
- make_from_usage_option;
- managed_design_object;
- manifold_constraining_context_dependent_shape_representation;
- manifold_subsurface_shape_representation;
- manifold_surface_shape_representation;
- mapped_item;
- mass_measure_with_unit;
- material_designation;
- material_property;
- material_removal_component_shape_aspect;
- measure_qualification;
- measure_representation_item;

- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- mounting_restriction_area;
- mounting_restriction_volume;
- multi_layer_component_shape_aspect;
- multi_layer_material_removal_component_shape_aspect;
- multi_layer_stratum_feature;
- next_assembly_usage_occurrence;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- padstack_occurrence;
- padstack_occurrence_shape_aspect_relationship;
- parameter_assignment_representation;
- part_connected_terminals_definition;

- part_mounting_feature;
- part_template_definition;
- part_text_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_network;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- plated_cutout_edge_segment;
- plated_inter_stratum_feature;
- plated_interconnect_module_edge_segment;
- plated_passage;
- polygonal_area;
- power_measure_with_unit;
- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship;
- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship;

- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- primitive_2d;
- printed_component;
- printed_component_link;
- printed_connector_template_terminal_relationship;
- printed_part_cross_section_template_terminal;
- printed_part_template_connected_terminals_definition;
- printed_part_template_link;
- printed_part_template_material;
- printed_part_template_material_link;
- printed_part_template_terminal;
- probe_access_area;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;

- product_definition_with_associated_documents;
- product_material_composition_relationship;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- promissory_usage_occurrence;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- quantified_assembly_component_usage;
- ratio_measure_with_unit;
- rectangular_area;
- rectangular_array_placement_group_component_definition;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- representation_relationship_with_transformation;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- requirements_property_group;
- resistance_measure_with_unit;

- routed_printed_component;
- rule_definition;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- sequential_laminate_passage_based_fabrication_joint;
- serial_numbered_effectivity;
- shape_aspect;
- shape_aspect_relationship;
- shape_dimension_representation;
- shape_modification;
- shape_representation;
- shape_representation_relationship;
- shape_representation_relationship_with_transformation;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- standard_uncertainty;
- stratum;
- stratum_concept_relationship;
- stratum_feature;
- stratum_occurrence_specific_padstack_definition;

- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- stratum_type_independent_padstack_definition;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- template_material_cross_section_boundary;
- template_stratum_assignment;
- text_literal;
- thermal_component;
- thermal_feature;
- thermal_isolation_component_shape_aspect;
- thermal_isolation_removal_template_definition;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- trimmed_curve;
- type_qualifier;
- uncertainty_qualifier;
- usage_view_connection_zone_terminal_shape_relationship;
- vector;
- volume_measure_with_unit;
- volume_shape_intersection.

6.11 Geometric Dimensioning and Tolerancing

This data includes the 14 geometric tolerances in ISO 1101. Angularity, circularity, circular run-out, concentricity, cylindricity, flatness, parallelism, perpendicularity, position, profile of any line, profile of any surface, straightness, symmetry, and total run-out. This data includes dimension, limits and fits. This data includes datum system definition. This data includes the tolerance zones in ISO 1101 and ASME Y14.5.(e.g., cylindrical, parallelepiped, projected, and conical). Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- authorization;
- document;
- end_item_identification;
- geometric_dimensioning_and_tolerance;
- product_identification;
- shape.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/geometric_dimensioning_and_tolerancing_class.exp..

6.12 Product Rule

This data includes support for rule creation, management, assignment to product data or features, assignment to product or requirement parameters. Complete use of this capability requires use of external definition for the expressions. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;

- end_item_identification;
- external_definition;
- product_rule;
- source_control.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- applied_date_and_time_assignment;
- approval;
- area_measure_with_unit;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- celsius_temperature_measure_with_unit;
- certification;
- colour_rgb;
- compound_representation_item;

- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- descriptive_representation_item;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;

- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_unit;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_model_port_relationship;
- managed_design_object;
- mass_measure_with_unit;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;

- person_and_organization;
- physical_unit;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_with_specified_source;
- product_definition_with_associated_documents;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- representation;
- representation_item;
- representation_relationship;
- requirement_definition;
- resistance_measure_with_unit;
- rule_action;
- rule_complex_clause;
- rule_conclusion_definition;

- rule_definition;
- rule_function_definition;
- rule_premise_definition;
- rule_set;
- rule_simple_clause;
- scattering_parameter_measure_with_unit;
- security_classification;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;
- structured_text_representation_item;
- table_representation_item;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.13 Functional Decomposition

This data includes specification of the folded and unfolded (elaborated) hierarchical product definition in the functional view. Support is provided for a usage view and a design view. The ability to exchange data defining a functional test bench and functional specification based on signals is included. Signal definition relies on external definition, but signal properties may be represented. Support is provided for both lumped element and distributed port properties. Analysis models may be included in the exchange structure with close integration accomplished by pin mapping. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_with_nodal_representation;
- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- library_item;
- planned_characteristic;
- product_identification;
- signal;
- source_control.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- applied_group_assignment;
- approval;
- area_measure_with_unit;
- assembly_group_component_definition;
- assembly_module_terminal;
- bare_die_terminal;
- bond_category;
- bus_element_link;
- bus_structural_definition;
- calendar_date;
- capacitance_measure_with_unit;
- celsius_temperature_measure_with_unit;
- certification;
- characterized_product_category;

- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- composite_signal_property_relationship;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- descriptive_representation_item;
- design_make_from_relationship;
- document;
- document_identifier;
- document_usage_constraint;

- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electrical_network;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluated_characteristic;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- illuminance_measure_with_unit;

- inductance_measure_with_unit;
- inter_stratum_feature;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- laminate_component_interface_terminal;
- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_functional_unit_terminal_definition_relationship;
- make_from_model_port_relationship;
- managed_design_object;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;

- minimally_defined_bare_die_terminal;
- model_parameter;
- network_node_definition;
- organization;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_unit;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;

- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_part_template_link;
- printed_part_template_terminal;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_relationship;
- product_definition_with_associated_documents;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;

- representation_relationship;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;
- rule_definition;
- scalar_terminal_definition_link;
- scattering_parameter_measure_with_unit;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;
- signal;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- table_representation_item;
- thermal_feature;

- thermal_network;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.14 Package Functional and Physical Characterization

Package Data includes following information: Case style, material identification, environmental constraints, performance data, simulation models. Formal encapsulation of external data type definitions is made for parametric data and analysis models. Terminal identification is provided to ensure consistency between 2D and 3D geometric models. Package body material is included. Terminals may have core and surface materials. Shape, material, technology, tolerance on shape and on parameters, units are included. Extensive support for orientation synchronization between enterprises is provided. Shapes included may be categorized according to discipline (e.g, thermal analysis, vibration analysis), and analysis environment (e.g., assembly, end-user). Geometric Dimensioning and Tolerancing is included. Complex packages may be treated by the design owner as an interconnect product and publish the information in package as the customer view. Formal Reference information to a defining document is provided. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- altered_package;
- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- datum_based_characteristic;
- device_marking;
- document;
- end_item_identification;

- external_definition;
- library_item;
- material_product;
- package;
- part_feature;
- physical_unit_shape;
- planned_characteristic;
- product_identification;
- shape;
- source_control;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/package_functional_and_physical_characterization_class.exp..

6.15 Geometrically Bounded Surface Model

This class shall not be implemented by itself.

The class requires the implementation of the following Units of Functionality:

- geometrically_bounded_2d_wireframe;
- non_topological_surface.

The class requires the implementation of the following AIM entities:

- plane_angle_measure_with_unit;
- geometric_representation_context;
- geometric_representation_item;

- geometrically_bounded_2d_wireframe_representation;
- geometrically_bounded_surface_shape_representation;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- length_measure_with_unit;
- managed_design_object;
- measure_with_unit;
- shape_representation.

6.16 Wireframe Model With Topology

This class shall not be implemented by itself.

The class requires the implementation of the following Units of Functionality:

- wireframe_2d_model_with_topology;
- wireframe_with_topology.

The class requires the implementation of the following AIM entities:

- plane_angle_measure_with_unit;
- edge_based_2d_wireframe_shape_representation;
- edge_based_wireframe_shape_representation;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- length_measure_with_unit;
- managed_design_object;

- measure_with_unit;
- shape_representation;
- shell_based_2d_wireframe_shape_representation;
- shell_based_wireframe_shape_representation.

6.17 Advanced Boundary Representation

This class shall not be implemented by itself.

The class requires the implementation of the following Units of Functionality:

- advanced_boundary_representation.

The class requires the implementation of the following AIM entities:

- advanced_brep_shape_representation;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- length_measure_with_unit;
- managed_design_object;
- measure_with_unit;
- plane_angle_measure_with_unit;
- shape_representation.

6.18 Constructive Solid Geometry

This class shall not be implemented by itself.

The class requires the implementation of the following Units of Functionality:

- constructive_solid_geometry.

The class requires the implementation of the following AIM entities:

- plane_angle_measure_with_unit;
- block;
- boolean_result;
- csg_shape_representation;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- length_measure_with_unit;
- managed_design_object;
- measure_with_unit;
- primitive_2d;
- right_angular_wedge;
- right_circular_cylinder;
- shape_representation.

6.19 Extruded Solid

This class shall not be implemented by itself.

The class requires the implementation of the following Units of Functionality:

- solid_of_linear_extrusion.

The class requires the implementation of the following AIM entities:

- plane_angle_measure_with_unit;
- extruded_area_solid;

- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- length_measure_with_unit;
- managed_design_object;
- measure_with_unit;
- shape_representation.

6.20 Geometrically Bounded 2d Wireframe Model

This class shall not be implemented by itself.

The class requires the implementation of the following AIM entities:

- geometrically_bounded_2d_wireframe_representation.

6.21 Wireframe 2d Model With Topology

This class shall not be implemented by itself.

The class requires the implementation of the following AIM entities:

- edge_based_2d_wireframe_shape_representation;
- shell_based_2d_wireframe_shape_representation.

6.22 Curve 2d

This class shall not be implemented by itself.

The class requires the implementation of the following AIM entities:

- curve;
- polyline.

6.23 Basic Curve 2d

Includes Only Lines, Circle and Arc Subtype of Conic. This class shall not be implemented by itself.

The class requires the implementation of the following AIM entities:

- circle;
- composite_curve;
- line;
- polyline;
- trimmed_curve.

6.24 Laminate Assembly Design

This data provides physical assembly information sufficient to allow communication of the arrangement of laminates in an interconnect product, and required interconnections among the materials assembled. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- authorization;
- characteristic;
- datum_based_characteristic;
- end_item_identification;
- laminate_assembly_design;
- library_item;
- product_identification;
- sequential_laminate_assembly_design.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;

- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_representation;
- applied_promissory_usage_in_product_model_assignment;
- approval;
- area_measure_with_unit;
- assembly_component_usage;
- assembly_group_component_definition;
- assembly_joint;
- assembly_module_terminal;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die_terminal;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- characterized_product_category;
- colour_rgb;
- component_definition;
- component_feature;

- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_mating_constraint_condition;
- component_mounting_feature;
- component_terminal;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- descriptive_representation_item;
- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;

- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_item;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- interconnect_module_component_surface_feature;
- interconnect_module_stratum_based_terminal;
- interconnect_module_terminal;
- interface_component_definition;
- laminate_component_interface_terminal;

- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- managed_design_object;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- next_assembly_usage_occurrence;
- organization;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_part_terminal;
- padstack_definition;

- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_with_associated_documents;
- product_related_characterized_product_category;

- product_specific_parameter_value_assignment;
- promissory_usage_occurrence;
- property_definition;
- property_definition_representation;
- qualitative_uncertainty;
- quantified_assembly_component_usage;
- ratio_measure_with_unit;
- representation;
- representation_item;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;
- rule_definition;
- scattering_parameter_measure_with_unit;
- secondary_orientation_feature;
- security_classification;
- sequential_laminate_passage_based_fabrication_joint;
- shape_aspect;
- shape_aspect_relationship;
- signal;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;

- structured_text_representation_item;
- table_representation_item;
- thermal_feature;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.25 Connection Zone Based Model Extraction

This data provides information sufficient to allow communication of more detailed connection points of analysis models than is available using feature assignment to analytical model ports. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- characteristic;
- connection_zone_to_analytical_model_port_mapping;
- document;
- end_item_identification;
- external_definition.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;

- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- approval;
- area_measure_with_unit;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- celsius_temperature_measure_with_unit;
- colour_rgb;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- descriptive_representation_item;
- document;
- document_identifier;
- document_usage_constraint;

- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_unit;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;

- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_model_port_relationship;
- managed_design_object;
- mass_measure_with_unit;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- product;
- product_concept;
- product_definition;

- product_definition_formation;
- product_definition_with_associated_documents;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- representation;
- representation_item;
- representation_relationship;
- requirement_definition;
- resistance_measure_with_unit;
- rule_definition;
- scattering_parameter_measure_with_unit;
- security_classification;
- shape_aspect;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;
- structured_text_representation_item;
- table_representation_item;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.26 Functional Specification

This data provides information sufficient to allow communication of the behavioural specification of product functions. Formal encapsulation of external data type definitions is made for parametric data and signal characterization. Configuration management information and design management information is provided.

The class requires the implementation of the following Units of Functionality:

- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_specification;
- functional_usage_view;
- product_identification;
- signal;
- source_control.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_representation;
- applied_group_assignment;
- approval;
- area_measure_with_unit;

- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- celsius_temperature_measure_with_unit;
- certification;
- characterized_product_category;
- colour_rgb;
- component_functional_terminal;
- component_functional_unit;
- composite_signal_property_relationship;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- descriptive_representation_item;
- design_make_from_relationship;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;

- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- length_measure_with_unit;
- library_defined_functional_unit;

- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_functional_unit_terminal_definition_relationship;
- managed_design_object;
- mass_measure_with_unit;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;

- product;
- product_category_relationship;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_formation_relationship;
- product_definition_formation_with_specified_source;
- product_definition_with_associated_documents;
- product_related_characterized_product_category;
- product_specific_parameter_value_assignment;
- property_definition;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- representation;
- representation_item;
- requirement_definition;
- resistance_measure_with_unit;
- rule_definition;
- scalar_terminal_definition_link;
- scattering_parameter_measure_with_unit;
- security_classification;
- shape_aspect;
- signal;

- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;
- structured_text_representation_item;
- table_representation_item;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.27 Physical Unit Physical Characterization

Physical Unit Physical Characterization data includes following information: Material identification, environmental constraints, performance data, simulation models. Formal encapsulation of external data type definitions is made for parametric data and analysis models. Shape, material, technology, tolerance on shape and on parameters, units are included. Shapes included may be categorized according to discipline (e.g, thermal analysis, vibration analysis), and analysis environment (e.g., assembly, end-user). Geometric Dimensioning and Tolerancing is included. A Formal Reference capability to the defining document is provided. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- authorization;
- characteristic;
- datum_based_characteristic;
- device_marking;
- document;
- end_item_identification;
- external_definition;

- library_item;
- material_product;
- package;
- part_feature;
- physical_unit;
- physical_unit_shape;
- planned_characteristic;
- product_identification;
- shape;
- source_control;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/physical_unit_physical_characterization_class.exp..

6.28 Packaged Part White Box Model

Packaged Part White Box Model Data includes following information: device model, package data, functional data, environmental constraints, performance data, simulation models. Assembly arrangement of devices included in the package to compose the packaged part is explicitly provided. Mapping of analysis model connection points to package terminals is provided. Formal encapsulation of external data type definitions is made for parametric data and analysis models. Pin mapping is provided to ensure consistency between device views. Shape, material, technology, tolerance on shape and on parameters, units are included. Shapes included may be categorized according to discipline (e.g. thermal analysis, vibration analysis), and analysis environment (e.g., design, assembly, end-user). Geometric Dimensioning and Tolerancing is included. The functional data includes two distinct data sets: hierarchical functional decomposition into behavioural elements embedded in a network represented at each level by a nodal formulation; collection of device terminals (e.g., connector terminals, printed circuit board interface areas, jumper ends) that implement the functionality of a network node. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- altered_part;
- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_with_nodal_representation;
- functional_definition_to_physical_unit_usage_view_mapping;
- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;

- library_item;
- material_product;
- package;
- packaged_part_terminal_analytical_representation;
- packaged_part_white_box_model;
- part_feature;
- part_feature_to_analytical_model_port_mapping;
- physical_unit;
- physical_unit_analytical_representation;
- physical_unit_shape;
- product_identification;
- shape;
- signal;
- source_control.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/packaged_part_white_box_model_class.exp..

6.29 Printed Part Functional and Physical Characterization

Printed Part Data includes following information: device model, layout template data, functional data, environmental constraints, performance data, simulation models. Mapping of analysis model connection points to printed part terminals is provided. Formal encapsulation of external data type definitions is made for parametric data and analysis models. Pin mapping is provided to ensure consistency between device views. Shape, material, technology, tolerance on shape and on parameters, units are included. Shapes included may be categorized according to discipline (e.g. thermal analysis, vibration analysis), and analysis environment (e.g., design, assembly, end-user). Geometric Dimensioning and Tolerancing is included. The functional data includes two distinct data sets: hierarchical functional decomposition into behavioural elements embedded in a network represented at each level by a nodal formulation; collection of device terminals that implement the functionality of a network node. Configuration management information and design management information is provided. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- functional_decomposition_with_nodal_representation;
- functional_definition_to_printed_part_mapping;
- functional_specification;
- functional_unit_analytical_representation;
- functional_usage_view;
- functional_usage_view_terminal_to_analytical_model_port_mapping;
- library_item;
- material_product;
- part_template;

- part_template_shape;
- printed_part_template;
- product_identification;
- shape;
- signal;
- source_control;
- styled_curve;
- styled_text.

The class requires the implementation of the following AIM entities:

- ERROR: Could not open schemas/classes/printed_part_functional_and_physical_characterization_class.exp..

6.30 Open Shell Model

This class shall not be implemented by itself.

The class requires the implementation of the following AIM entities:

- manifold_surface_shape_representation.

6.31 Assembly Functional Interface Requirements

Assembly Functional Interface Requirements includes the following information: functional interface model, signal constraints, external definition support, Product identification, functional usage view. Support for documentation is provided.

The class requires the implementation of the following Units of Functionality:

- assembly_functional_interface_requirement;
- authorization;
- characteristic;
- document;

- end_item_identification;
- external_definition;
- functional_usage_view.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_representation;
- applied_group_assignment;
- approval;
- area_measure_with_unit;
- assembly_module_terminal;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die_terminal;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- colour_rgb;

- component_definition;
- component_functional_terminal;
- component_functional_unit;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- design_make_from_relationship;
- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;

- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;

- global_unit_assigned_context;
- group_shape_aspect;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- interconnect_module_terminal;
- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_functional_unit_terminal_definition_relationship;
- managed_design_object;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;

- minimally_defined_bare_die_terminal;
- model_parameter;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;

- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_part_template_link;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_with_associated_documents;
- property_definition;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- representation;
- representation_item;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;
- rule_definition;
- scalar_terminal_definition_link;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;

- shape_aspect;
- shape_aspect_relationship;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- stratum;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- table_representation_item;
- thermal_feature;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.32 Assembly Physical Interface Requirements

Assembly Physical Interface Requirements includes the following information: critical component placement data, critical component placement or keepout or keepin requirements, connector placement data, mating connector placement data, geometric interface (fit) requirements to next level assembly. Supporting data including external definition, properties, product identification, documentation and authorization are included.

The class requires the implementation of the following Units of Functionality:

- assembly_component_placement_requirements;
- assembly_interface_component;
- assembly_physical_interface_requirement;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_representation;
- approval;
- area_measure_with_unit;
- assembly_group_component_definition;

- assembly_module_terminal;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die_terminal;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_mounting_feature;
- component_terminal;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;

- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;

- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- group_product_definition;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- interface_component_definition;
- laminate_component_interface_terminal;
- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;

- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- managed_design_object;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- mounting_restriction_area;
- mounting_restriction_volume;
- organization;
- package;
- package_body;

- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;

- printed_part_template_link;
- printed_part_template_terminal;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;
- rule_definition;
- scattering_parameter_measure_with_unit;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;

- shape_aspect_relationship;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- specified_higher_usage_occurrence;
- standard_uncertainty;
- stratum;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- table_representation_item;
- thermal_feature;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.33 Assembly Requirement Allocation

Assembly Requirement Allocation includes the following information: complete functional and physical interface requirements, complete capability to allocate enterprise or external agency requirements to assembly function or physical design, model based requirement description using simulation models and models with assigned parametric data. Supporting data including document, authorization, product identification, external definition, and authorization are included.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- assembly_functional_interface_requirement;
- assembly_functional_requirement_allocation;
- assembly_physical_interface_requirement;
- assembly_physical_requirement_allocation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;

- analytical_model_port;
- analytical_representation;
- approval;
- area_measure_with_unit;
- assembly_group_component_definition;
- assembly_joint;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;

- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- direction;
- discrete_shield;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;

- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_bare_die;
- externally_defined_functional_unit;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;

- global_unit_assigned_context;
- group_shape_aspect;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- interface_component_definition;
- interface_mounted_join;
- laminate_component_interface_terminal;
- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;
- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;

- magnetic_flux_measure_with_unit;
- make_from_model_port_relationship;
- managed_design_object;
- mapped_item;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- mounting_restriction_area;
- mounting_restriction_volume;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;
- packaged_connector;
- packaged_part;
- packaged_part_terminal;

- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_node_requirement_to_implementing_component_allocation;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_part_template_link;
- printed_part_template_terminal;
- product;

- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;
- rule_definition;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;

- shape_aspect_relationship;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- specified_higher_usage_occurrence;
- standard_uncertainty;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- table_representation_item;
- thermal_component;
- thermal_feature;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.34 Interconnect Requirement Allocation

Interconnect Requirement Allocation includes the following information: complete functional and physical requirements, complete capability to allocate enterprise or external agency requirements to interconnect function or physical design, model based requirement description using simulation models and models with assigned parametric data. Capability to explicitly describe the required external connections for a printed wiring board or simple substrate. Capability to segregate the required external connections from the required functions other than simple interconnect for printed circuit boards (including printed components). Supporting data including document, authorization, product identification, external definition, and authorization are included.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- analytical_representation;
- authorization;
- characteristic;
- document;
- end_item_identification;
- external_definition;
- interconnect_functional_requirement_allocation;
- interconnect_physical_requirement_allocation.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;

- analytical_representation;
- applied_document_reference;
- approval;
- area_measure_with_unit;
- assembly_group_component_definition;
- assembly_joint;
- assembly_module_terminal;
- assembly_module_usage_view_connector_relationship;
- axis2_placement_2d;
- axis2_placement_3d;
- bare_die;
- bare_die_terminal;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- cartesian_point;
- celsius_temperature_measure_with_unit;
- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;

- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- connection_zone_interface_plane_relationship;
- contract;
- coordinated_representation_item;
- curve_style;
- curve_style_font;
- curve_style_font_and_scaling;
- data_set_representation_item;
- date_and_time;
- datum_reference_frame;
- datum_system;
- descriptive_representation_item;
- direction;
- discrete_shield;
- document;
- document_identifier;
- document_usage_constraint;

- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;
- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- evaluation_product_definition;
- expanded_uncertainty;
- external_definition;
- externally_defined_bare_die;
- externally_defined_curve_font;
- externally_defined_functional_unit;
- externally_defined_package;
- externally_defined_packaged_connector;
- externally_defined_packaged_part;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- functional_specification;
- functional_specification_definition;
- functional_unit;

- functional_unit_terminal_definition;
- geometric_representation_context;
- geometric_representation_item;
- global_uncertainty_assigned_context;
- global_unit_assigned_context;
- grouped_requirements_property;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- inter_stratum_feature;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- interface_component_definition;
- interface_mounted_join;
- join_shape_aspect;
- laminate_component_interface_terminal;
- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;
- library_defined_bare_die;
- library_defined_functional_unit;
- library_defined_package;
- library_defined_packaged_connector;
- library_defined_packaged_part;

- library_defined_physical_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- make_from_model_port_relationship;
- managed_design_object;
- mapped_item;
- mass_measure_with_unit;
- material_designation;
- material_property;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- network_node_definition;
- organization;
- package;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_component;

- packaged_connector;
- packaged_part;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_connected_terminals_definition;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;
- physical_network;
- physical_node_requirement_to_implementing_component_allocation;
- physical_unit;
- physical_unit_datum;
- physical_unit_datum_feature;
- physical_unit_network_definition;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;

- pressure_measure_with_unit;
- primary_orientation_feature;
- printed_part_template_link;
- printed_part_template_terminal;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- property_definition_relationship;
- property_definition_representation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- representation_relationship;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- resistance_measure_with_unit;

- rule_definition;
- scattering_parameter_measure_with_unit;
- seating_plane;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;
- shape_modification;
- shape_representation;
- signal;
- solid_angle_measure_with_unit;
- solid_curve_font;
- standard_uncertainty;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- styled_item;
- table_representation_item;
- thermal_component;
- thermal_feature;

- thermal_isolation_removal_template_definition;
- thermodynamic_temperature_measure_with_unit;
- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit;
- volume_shape_intersection.

6.35 Functional Requirement Allocation

Functional Requirement Allocation includes the following information: allocation of customer, regulatory agency, or enterprise requirements to a functional element or to a functional specification for various phases and purposes of a design process.

The class requires the implementation of the following Units of Functionality:

- functional_requirement_allocation.

The class requires the implementation of the following AIM entities:

- analytical_model;
- analytical_model_definition;
- analytical_representation;
- approval;
- bond_category;
- calendar_date;
- component_functional_terminal;
- component_functional_unit;
- compound_representation_item;
- contract;

- coordinated_representation_item;
- date_and_time;
- descriptive_representation_item;
- document;
- document_identifier;
- document_usage_constraint;
- ee_specification;
- evaluation_product_definition;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- library_defined_functional_unit;
- library_defined_physical_unit;
- managed_design_object;
- material_designation;
- measure_representation_item;
- measure_with_unit;

- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_network_definition;
- product;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- representation;
- representation_item;
- requirement_definition;
- requirements_property;
- rule_definition;
- security_classification;
- signal;
- stratum;
- structured_text_representation_item;
- table_representation_item.

6.36 Product Requirement Allocation

Product Requirement Allocation includes the following information: allocation data describing the assignment of customer, regulatory agency, and enterprise requirements to a product. These requirements can be assigned to products or configuration items or product versions in various phases of the design process.

The class requires the implementation of the following Units of Functionality:

- product_requirement_allocation.

The class requires the implementation of the following AIM entities:

- analytical_model;
- analytical_model_definition;
- analytical_representation;
- approval;
- bond_category;
- calendar_date;
- component_functional_terminal;
- component_functional_unit;
- compound_representation_item;
- configuration_item;
- contract;
- coordinated_representation_item;
- date_and_time;
- descriptive_representation_item;
- document;
- document_identifier;
- document_usage_constraint;

- ee_specification;
- evaluation_product_definition;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- library_defined_functional_unit;
- library_defined_physical_unit;
- managed_design_object;
- material_designation;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;
- person_and_organization;

- physical_unit;
- physical_unit_network_definition;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- representation;
- representation_item;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- rule_definition;
- security_classification;
- shape_aspect_relationship;
- signal;
- stratum;
- structured_text_representation_item;
- table_representation_item.

6.37 Datum Difference Based Model Definition

Datum Difference Based Model Definition includes the following information: data that describes interfaces to simulation models where the interfaces are based on Datums. Base geometric models may be created that Datum system and baseline parameters, which are then fed to a simulator, along with instance data from the product model, which is also correlated with the base geometric model. A Functional usage view is supported, providing the capability to map the model onto the functional definition for simulation using circuit level simulation. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- analytical_model;
- authorization;
- datum_difference_based_model;
- document;
- end_item_identification;
- external_definition;
- functional_usage_view.

The class requires the implementation of the following AIM entities:

- analytical_model;
- analytical_model_definition;
- analytical_model_make_from_relationship;
- analytical_model_port;
- analytical_representation;
- applied_group_assignment;
- approval;
- bond_category;
- calendar_date;
- configuration_item;

- contract;
- coordinated_representation_item;
- date_and_time;
- datum_difference;
- datum_difference_analytical_model_port_assignment;
- datum_difference_based_model_parameter;
- datum_difference_functional_unit_usage_view_terminal_assignment;
- descriptive_representation_item;
- design_make_from_relationship;
- document;
- document_identifier;
- document_usage_constraint;
- ee_specification;
- evaluation_product_definition;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- functional_terminal_group;
- functional_unit;
- functional_unit_terminal_definition;
- library_defined_functional_unit;

- library_defined_physical_unit;
- make_from_functional_unit_terminal_definition_relationship;
- make_from_model_port_relationship;
- managed_design_object;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- padstack_definition;
- parameter_assignment_representation;
- person;
- person_and_organization;
- physical_unit;
- physical_unit_datum;
- physical_unit_network_definition;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_with_associated_documents;
- representation_item;
- representation_relationship;
- requirement_definition;

- rule_definition;
- scalar_terminal_definition_link;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;
- stratum;
- structured_text_representation_item.

6.38 Datum Difference Based Characteristic

Datum Difference Based Characteristic includes the following information: data that describes parameters for a physical unit, a template, or a model, where the characteristic is based on a Datum, or where the parameter is based on the difference between two Datum. Supporting data including document, product identification, and external definition are included. This data includes at least one geometric representation.

The class requires the implementation of the following Units of Functionality:

- characteristic;
- datum_based_characteristic;
- datum_difference_based_characteristic;
- document;
- end_item_identification;
- external_definition.

The class requires the implementation of the following AIM entities:

- absorbed_dose_measure_with_unit;
- activity_measure_with_unit;
- amount_of_substance_measure_with_unit;
- approval;

- area_measure_with_unit;
- bond_category;
- calendar_date;
- capacitance_measure_with_unit;
- celsius_temperature_measure_with_unit;
- colour_rgb;
- compound_representation_item;
- conductance_measure_with_unit;
- configuration_item;
- contract;
- coordinated_representation_item;
- data_set_representation_item;
- date_and_time;
- datum_difference;
- datum_difference_based_characteristic;
- descriptive_representation_item;
- direction;
- document;
- document_identifier;
- document_usage_constraint;
- dose_equivalent_measure_with_unit;
- ee_specification;
- electric_charge_measure_with_unit;

- electric_current_measure_with_unit;
- electromotive_force_measure_with_unit;
- energy_measure_with_unit;
- expanded_uncertainty;
- external_definition;
- force_measure_with_unit;
- frequency_measure_with_unit;
- illuminance_measure_with_unit;
- inductance_measure_with_unit;
- length_measure_with_unit;
- luminous_flux_measure_with_unit;
- luminous_intensity_measure_with_unit;
- magnetic_flux_density_measure_with_unit;
- magnetic_flux_measure_with_unit;
- managed_design_object;
- mass_measure_with_unit;
- measure_qualification;
- measure_representation_item;
- measure_with_unit;
- model_parameter;
- organization;
- parameter_assignment_representation;
- person;

- person_and_organization;
- physical_unit_datum;
- plane_angle_measure_with_unit;
- power_measure_with_unit;
- precision_qualifier;
- pressure_measure_with_unit;
- product;
- product_concept;
- product_definition_formation;
- qualitative_uncertainty;
- ratio_measure_with_unit;
- representation;
- representation_item;
- resistance_measure_with_unit;
- scattering_parameter_measure_with_unit;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;
- solid_angle_measure_with_unit;
- standard_uncertainty;
- structured_text_representation_item;
- table_representation_item;
- thermodynamic_temperature_measure_with_unit;

- time_measure_with_unit;
- type_qualifier;
- uncertainty_qualifier;
- volume_measure_with_unit.

6.39 Design Management

This data includes configuration management and design activity control and includes change request and work request.

The class requires the implementation of the following Units of Functionality:

- design_activity_control;
- design_management.

The class requires the implementation of the following AIM entities:

- add_design_object_assignment;
- add_design_object_request_assignment;
- analytical_model;
- analytical_model_definition;
- analytical_representation;
- approval;
- assembly_group_component_definition;
- assembly_module_terminal;
- bare_die_terminal;
- bond_category;
- calendar_date;
- change_from_design_object_assignment;

- change_from_design_object_request_assignment;
- colour_rgb;
- component_definition;
- component_feature;
- component_functional_terminal;
- component_functional_unit;
- component_interface_terminal;
- component_mounting_feature;
- component_shape_aspect;
- component_terminal;
- composite_shape_aspect;
- compound_representation_item;
- configuration_item;
- contract;
- coordinated_representation_item;
- date_and_time;
- delete_design_object_assignment;
- delete_design_object_request_assignment;
- descriptive_representation_item;
- directed_action;
- document;
- document_identifier;
- document_usage_constraint;

- ee_specification;
- evaluated_characteristic;
- evaluation_product_definition;
- external_definition;
- externally_defined_functional_unit;
- externally_defined_physical_unit;
- externally_defined_product_definition;
- footprint_definition;
- frozen_assigned_item;
- functional_specification;
- functional_specification_definition;
- functional_unit;
- functional_unit_terminal_definition;
- inter_stratum_feature;
- interconnect_module_component_surface_feature;
- interconnect_module_terminal;
- laminate_component_interface_terminal;
- land_physical_template;
- land_template_terminal;
- length_measure_with_unit;
- library_defined_functional_unit;
- library_defined_physical_unit;
- managed_design_object;

- material_designation;
- material_property;
- measure_representation_item;
- measure_with_unit;
- minimally_defined_bare_die_terminal;
- model_parameter;
- organization;
- owner_assigned_item;
- owner_assignment;
- package_body;
- package_body_surface;
- package_terminal;
- packaged_part_terminal;
- padstack_definition;
- parameter_assignment_representation;
- part_mounting_feature;
- part_template_definition;
- part_tooling_feature;
- passage_technology;
- person;
- person_and_organization;
- physical_connectivity_definition;
- physical_connectivity_element;

- physical_unit;
- physical_unit_network_definition;
- primary_orientation_feature;
- printed_part_template_link;
- printed_part_template_terminal;
- product;
- product_concept;
- product_definition;
- product_definition_formation;
- product_definition_relationship;
- product_definition_with_associated_documents;
- property_definition;
- reference_packaged_part_interconnect_implementation;
- representation;
- representation_item;
- requirement_allocation_group;
- requirement_definition;
- requirements_property;
- rule_definition;
- secondary_orientation_feature;
- security_classification;
- shape_aspect;
- shape_aspect_relationship;

- signal;
- stratum;
- stratum_feature;
- stratum_surface;
- stratum_technology;
- stratum_technology_link;
- structured_text_representation_item;
- table_representation_item;
- thermal_feature;
- versioned_action_request.

NOTE ISO 10303-100¹⁾ defines the abstract test suite to be used in the assessment of conformance. ISO 10303-32²⁾ describes the conformance assessment process.

¹⁾To be published

²⁾To be published

Annex A (normative) AIM EXPRESS expanded listing

The following EXPRESS is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used. *)

```
(*
Created:
Wed Aug 30 22:55:21 EDT 2000
*)
```

```
SCHEMA electronic_assembly_interconnect_and_packaging_design;
```

```
CONSTANT
dummy_gri : geometric_representation_item := representation_item('') ||
           geometric_representation_item();
END_CONSTANT;
```

```
TYPE action_assigned_item = SELECT
  (product_definition_formation,
   rule_definition);
END_TYPE; -- action_assigned_item
```

```
TYPE ahead_or_behind = ENUMERATION OF
  (ahead,
   behind);
END_TYPE; -- ahead_or_behind
```

```
TYPE amount_of_substance_measure = REAL;
END_TYPE; -- amount_of_substance_measure
```

```
TYPE angle_relator = ENUMERATION OF
  (equal,
   large,
   small);
END_TYPE; -- angle_relator
```

```
TYPE approval_assigned_item = SELECT
  (action,
   action_directive,
   alternate_product_relationship,
   applied_date_and_time_assignment,
   applied_document_reference,
   certification,
```

change_request,
characterized_object,
colour,
component_definition,
component_shape_aspect,
composite_curve_segment,
configuration_effectivity,
configuration_item,
contract,
curve_style,
curve_style_font,
curve_style_font_and_scaling,
date,
dimension_related_tolerance_zone_element,
dimensional_characteristic_representation,
dimensional_size,
directed_action,
document,
document_usage_constraint,
externally_defined_item,
fill_area_style,
fill_area_style_colour,
geometric_tolerance,
geometric_tolerance_relationship,
group,
inter_stratum_feature,
material_designation,
measure_with_unit,
plus_minus_tolerance,
pre_defined_item,
product,
product_concept,
product_definition,
product_definition_formation,
product_definition_formation_relationship,
product_definition_formation_with_specified_source,
product_definition_relationship,
property_definition,
property_definition_relationship,
property_definition_representation,
representation,
representation_context,
representation_item,
representation_relationship,
runout_zone_orientation,
security_classification,
shape_aspect,
shape_aspect_relationship,
start_request,
stratum_feature,
tolerance_value,
tolerance_zone_form,

```

        versioned_action_request);
END_TYPE; -- approval_assigned_item

```

```

TYPE area_measure = REAL;
END_TYPE; -- area_measure

```

```

TYPE axis2_placement = SELECT
    (axis2_placement_2d,
     axis2_placement_3d);
END_TYPE; -- axis2_placement

```

```

TYPE b_spline_curve_form = ENUMERATION OF
    (polyline_form,
     circular_arc,
     elliptic_arc,
     parabolic_arc,
     hyperbolic_arc,
     unspecified);
END_TYPE; -- b_spline_curve_form

```

```

TYPE b_spline_surface_form = ENUMERATION OF
    (plane_surf,
     cylindrical_surf,
     conical_surf,
     spherical_surf,
     toroidal_surf,
     surf_of_revolution,
     ruled_surf,
     generalised_cone,
     quadric_surf,
     surf_of_linear_extrusion,
     unspecified);
END_TYPE; -- b_spline_surface_form

```

```

TYPE boolean_operand = SELECT
    (solid_model,
     half_space_solid,
     csg_primitive,
     boolean_result);
END_TYPE; -- boolean_operand

```

```

TYPE boolean_operator = ENUMERATION OF
    (union,
     intersection,

```

```

        difference);
END_TYPE; -- boolean_operator

```

```

TYPE box_characteristic_select = SELECT
    (box_height,
     box_width,
     box_slant_angle,
     box_rotate_angle);
END_TYPE; -- box_characteristic_select

```

```

TYPE box_height = positive_ratio_measure;
END_TYPE; -- box_height

```

```

TYPE box_rotate_angle = plane_angle_measure;
END_TYPE; -- box_rotate_angle

```

```

TYPE box_slant_angle = plane_angle_measure;
END_TYPE; -- box_slant_angle

```

```

TYPE box_width = positive_ratio_measure;
END_TYPE; -- box_width

```

```

TYPE certification_assigned_item = SELECT
    (supplied_part_relationship,
     make_from_usage_option,
     product_definition_formation_with_specified_source);
END_TYPE; -- certification_assigned_item

```

```

TYPE change_request_item = SELECT
    (product_definition_formation,
     representation_relationship);
END_TYPE; -- change_request_item

```

```

TYPE character_spacing_select = SELECT
    (length_measure,
     ratio_measure,
     measure_with_unit,
     descriptive_measure);
END_TYPE; -- character_spacing_select

```

```
TYPE character_style_select = SELECT
    (character_glyph_style_stroke,
     character_glyph_style_outline,
     text_style_for_defined_font);
END_TYPE; -- character_style_select
```

```
TYPE characterized_definition = SELECT
    (characterized_object,
     characterized_product_definition,
     shape_definition);
END_TYPE; -- characterized_definition
```

```
TYPE characterized_material_property = SELECT
    (material_property_representation,
     product_material_composition_relationship);
END_TYPE; -- characterized_material_property
```

```
TYPE characterized_product_definition = SELECT
    (product_definition,
     product_definition_relationship);
END_TYPE; -- characterized_product_definition
```

```
TYPE classification_assigned_item = SELECT
    (requirements_property,
     signal,
     stratum);
END_TYPE; -- classification_assigned_item
```

```
TYPE context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure
```

```
TYPE contract_assigned_item = SELECT
    (alternate_product_relationship,
     directed_action,
     product,
     product_definition_formation);
END_TYPE; -- contract_assigned_item
```

```
TYPE count_measure = NUMBER;
END_TYPE; -- count_measure
```



```

TYPE csg_primitive = SELECT
    (sphere,
     block,
     right_angular_wedge,
     torus,
     right_circular_cone,
     right_circular_cylinder);
END_TYPE; -- csg_primitive

```

```

TYPE csg_select = SELECT
    (boolean_result,
     csg_primitive);
END_TYPE; -- csg_select

```

```

TYPE curve_font_or_scaled_curve_font_select = SELECT
    (curve_style_font_select,
     curve_style_font_and_scaling);
END_TYPE; -- curve_font_or_scaled_curve_font_select

```

```

TYPE curve_on_surface = SELECT
    (pcurve,
     surface_curve,
     composite_curve_on_surface);
END_TYPE; -- curve_on_surface

```

```

TYPE curve_or_annotation_curve_occurrence = SELECT
    (curve,
     annotation_curve_occurrence);
END_TYPE; -- curve_or_annotation_curve_occurrence

```

```

TYPE curve_or_render = SELECT
    (curve_style);
END_TYPE; -- curve_or_render

```

```

TYPE curve_style_font_select = SELECT
    (curve_style_font,
     pre_defined_curve_font,
     externally_defined_curve_font);
END_TYPE; -- curve_style_font_select

```

```

TYPE date_and_time_assigned_item = SELECT

```

```

    (approval_person_organization,
    certification,
    change,
    contract,
    directed_action,
    document,
    product_definition,
    rule_action,
    security_classification,
    versioned_action_request);
END_TYPE; -- date_and_time_assigned_item

```

```

TYPE date_assigned_item = SELECT
    (product_definition,
    versioned_action_request,
    directed_action,
    approval_person_organization,
    contract,
    security_classification,
    certification);
END_TYPE; -- date_assigned_item

```

```

TYPE date_time_select = SELECT
    (date,
    local_time,
    date_and_time);
END_TYPE; -- date_time_select

```

```

TYPE day_in_month_number = INTEGER;
END_TYPE; -- day_in_month_number

```

```

TYPE defined_symbol_select = SELECT
    (externally_defined_symbol);
END_TYPE; -- defined_symbol_select

```

```

TYPE descriptive_measure = STRING;
END_TYPE; -- descriptive_measure

```

```

TYPE dimension_count = INTEGER;
WHERE
    WR1: (SELF > 0);
END_TYPE; -- dimension_count

```

```

TYPE dimensional_characteristic = SELECT
  (dimensional_location,
   dimensional_size);
END_TYPE; -- dimensional_characteristic

```

```

TYPE document_assigned_item = SELECT
  (action_method,
   assembly_component_usage,
   bond_category,
   characterized_object,
   component_definition,
   component_interface_terminal,
   component_shape_aspect,
   component_terminal,
   configuration_item,
   descriptive_representation_item,
   externally_defined_item,
   functional_unit_terminal_definition,
   layer,
   material_designation,
   measure_representation_item,
   package_terminal,
   part_template_definition,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   property_definition,
   representation,
   representation_item,
   requirements_property,
   rule_definition,
   shape_aspect,
   signal,
   stratum);
END_TYPE; -- document_assigned_item

```

```

TYPE electric_current_measure = REAL;
END_TYPE; -- electric_current_measure

```

```

TYPE fill_area_style_tile_shape_select = SELECT
  (fill_area_style_tile_curve_with_style,
   fill_area_style_tile_coloured_region,
   fill_area_style_tile_symbol_with_style,
   externally_defined_tile);
END_TYPE; -- fill_area_style_tile_shape_select

```

```

TYPE fill_style_select = SELECT
    (fill_area_style_colour,
     externally_defined_tile_style,
     fill_area_style_tiles,
     externally_defined_hatch_style,
     fill_area_style_hatching);
END_TYPE; -- fill_style_select

```

```

TYPE font_select = SELECT
    (externally_defined_text_font);
END_TYPE; -- font_select

```

```

TYPE founded_item_select = SELECT
    (founded_item,
     representation_item);
END_TYPE; -- founded_item_select

```

```

TYPE geometric_set_select = SELECT
    (point,
     curve,
     surface);
END_TYPE; -- geometric_set_select

```

```

TYPE group_assigned_item = SELECT
    (component_definition,
     component_interface_terminal,
     component_shape_aspect,
     component_terminal,
     configuration_item,
     inter_stratum_feature,
     printed_part_template_terminal,
     product,
     product_definition_formation,
     representation,
     representation_item,
     requirements_property,
     rule_definition,
     shape_aspect,
     stratum,
     stratum_feature,
     stratum_surface);
END_TYPE; -- group_assigned_item

```

```

TYPE hiding_or_blanking_select = SELECT

```

```

        (annotation_fill_area,
         character_glyph_symbol_outline);
END_TYPE; -- hiding_or_blanking_select

```

```

TYPE hour_in_day = INTEGER;
WHERE
    WR1: ((0 <= SELF) AND (SELF < 24));
END_TYPE; -- hour_in_day

```

```

TYPE identifier = STRING;
END_TYPE; -- identifier

```

```

TYPE invisibility_context = SELECT
    (presentation_representation);
END_TYPE; -- invisibility_context

```

```

TYPE invisible_item = SELECT
    (styled_item,
     representation);
END_TYPE; -- invisible_item

```

```

TYPE knot_type = ENUMERATION OF
    (uniform_knots,
     unspecified,
     quasi_uniform_knots,
     piecewise_bezier_knots);
END_TYPE; -- knot_type

```

```

TYPE label = STRING;
END_TYPE; -- label

```

```

TYPE layered_item = SELECT
    (presentation_representation,
     representation_item);
END_TYPE; -- layered_item

```

```

TYPE length_measure = REAL;
END_TYPE; -- length_measure

```

```

TYPE limit_condition = ENUMERATION OF
    (maximum_material_condition,
     least_material_condition,
     regardless_of_feature_size);
END_TYPE; -- limit_condition

```

```

TYPE list_of_reversible_topology_item = LIST [0:?] OF
    reversible_topology_item;
END_TYPE; -- list_of_reversible_topology_item

```

```

TYPE luminous_intensity_measure = REAL;
END_TYPE; -- luminous_intensity_measure

```

```

TYPE managed_design_object = SELECT
    (action,
     alternate_product_relationship,
     applied_approval_assignment,
     applied_date_and_time_assignment,
     applied_document_reference,
     approval,
     approval_status,
     certification,
     characterized_object,
     colour,
     composite_curve_segment,
     configuration_effectivity,
     configuration_item,
     contract,
     curve_style,
     curve_style_font,
     curve_style_font_and_scaling,
     date,
     dimension_related_tolerance_zone_element,
     dimensional_characteristic_representation,
     dimensional_size,
     document,
     document_usage_constraint,
     externally_defined_item,
     fill_area_style,
     fill_area_style_colour,
     geometric_tolerance,
     geometric_tolerance_relationship,
     group,
     layer_connection_point,
     layer_connection_point_link,
     material_designation,
     measure_with_unit,
     organization,

```

```

person,
person_and_organization,
plus_minus_tolerance,
pre_defined_item,
product,
product_concept,
product_definition,
product_definition_formation,
product_definition_formation_relationship,
product_definition_relationship,
property_definition,
property_definition_relationship,
property_definition_representation,
representation,
representation_context,
representation_item,
representation_relationship,
runout_zone_orientation,
security_classification,
shape_aspect,
shape_aspect_relationship,
tolerance_value,
tolerance_zone_form,
versioned_action_request);
END_TYPE; -- managed_design_object

```

```

TYPE mass_measure = REAL;
END_TYPE; -- mass_measure

```

```

TYPE measure_value = SELECT
(length_measure,
mass_measure,
time_measure,
electric_current_measure,
thermodynamic_temperature_measure,
amount_of_substance_measure,
luminous_intensity_measure,
plane_angle_measure,
solid_angle_measure,
area_measure,
volume_measure,
ratio_measure,
parameter_value,
numeric_measure,
context_dependent_measure,
descriptive_measure,
positive_length_measure,
positive_plane_angle_measure,
positive_ratio_measure,
count_measure);

```

```
END_TYPE; -- measure_value
```

```
TYPE minute_in_hour = INTEGER;
WHERE
  WR1: ((0 <= SELF) AND (SELF <= 59));
END_TYPE; -- minute_in_hour
```

```
TYPE month_in_year_number = INTEGER;
WHERE
  WR1: ((1 <= SELF) AND (SELF <= 12));
END_TYPE; -- month_in_year_number
```

```
TYPE numeric_measure = NUMBER;
END_TYPE; -- numeric_measure
```

```
TYPE organization_assigned_item = SELECT
  (configuration_item,
   contract,
   directed_action,
   document,
   versioned_action_request,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_with_specified_source,
   representation,
   security_classification);
END_TYPE; -- organization_assigned_item
```

```
TYPE parameter_value = REAL;
END_TYPE; -- parameter_value
```

```
TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE; -- pcurve_or_surface
```

```
TYPE person_and_organization_assigned_item = SELECT
  (configuration_item,
   contract,
   document,
   product,
```



```

    product_definition_formation,
    product_definition,
    rule_action,
    security_classification,
    versioned_action_request);
END_TYPE; -- person_and_organization_assigned_item

```

```

TYPE person_assigned_item = SELECT
    (change,
     document,
     security_classification);
END_TYPE; -- person_assigned_item

```

```

TYPE person_organization_select = SELECT
    (person,
     organization,
     person_and_organization);
END_TYPE; -- person_organization_select

```

```

TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure

```

```

TYPE positive_length_measure = length_measure;
WHERE
    WR1: (SELF > 0);
END_TYPE; -- positive_length_measure

```

```

TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
    WR1: (SELF > 0);
END_TYPE; -- positive_plane_angle_measure

```

```

TYPE positive_ratio_measure = ratio_measure;
WHERE
    WR1: (SELF > 0);
END_TYPE; -- positive_ratio_measure

```

```

TYPE preferred_surface_curve_representation = ENUMERATION OF
    (curve_3d,
     pcurve_s1,
     pcurve_s2);
END_TYPE; -- preferred_surface_curve_representation

```

```
TYPE presentable_text = STRING;  
END_TYPE; -- presentable_text
```

```
TYPE presentation_representation_select = SELECT  
  (presentation_representation);  
END_TYPE; -- presentation_representation_select
```

```
TYPE presentation_style_select = SELECT  
  (curve_style,  
   fill_area_style,  
   text_style);  
END_TYPE; -- presentation_style_select
```

```
TYPE ratio_measure = REAL;  
END_TYPE; -- ratio_measure
```

```
TYPE replacement_from_item = SELECT  
  (rule_definition);  
END_TYPE; -- replacement_from_item
```

```
TYPE replacement_to_item = SELECT  
  (rule_definition);  
END_TYPE; -- replacement_to_item
```

```
TYPE reversible_topology = SELECT  
  (reversible_topology_item,  
   list_of_reversible_topology_item,  
   set_of_reversible_topology_item);  
END_TYPE; -- reversible_topology
```

```
TYPE reversible_topology_item = SELECT  
  (edge,  
   path,  
   face,  
   face_bound,  
   closed_shell,  
   open_shell);  
END_TYPE; -- reversible_topology_item
```

```
TYPE second_in_minute = REAL;
```

WHERE

```
    WR1: ((0 <= SELF) AND (SELF < 60));
END_TYPE; -- second_in_minute
```

```
TYPE security_classification_assigned_item = SELECT
    (assembly_component_usage,
     document,
     make_from_usage_option,
     product_definition_formation,
     product_definition);
END_TYPE; -- security_classification_assigned_item
```

```
TYPE set_of_reversible_topology_item = SET [0:?] OF
    reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_item
```

```
TYPE shape_definition = SELECT
    (product_definition_shape,
     shape_aspect,
     shape_aspect_relationship);
END_TYPE; -- shape_definition
```

```
TYPE shell = SELECT
    (vertex_shell,
     wire_shell,
     open_shell,
     closed_shell);
END_TYPE; -- shell
```

```
TYPE si_prefix = ENUMERATION OF
    (exa,
     peta,
     tera,
     giga,
     mega,
     kilo,
     hecto,
     deca,
     deci,
     centi,
     milli,
     micro,
     nano,
     pico,
     femto,
```

```

        atto);
END_TYPE; -- si_prefix

```

```

TYPE si_unit_name = ENUMERATION OF
    (metre,
     gram,
     second,
     ampere,
     kelvin,
     mole,
     candela,
     radian,
     steradian,
     hertz,
     newton,
     pascal,
     joule,
     watt,
     coulomb,
     volt,
     farad,
     ohm,
     siemens,
     weber,
     tesla,
     henry,
     degree_celsius,
     lumen,
     lux,
     becquerel,
     gray,
     sievert);
END_TYPE; -- si_unit_name

```

```

TYPE size_select = SELECT
    (positive_length_measure,
     measure_with_unit,
     descriptive_measure);
END_TYPE; -- size_select

```

```

TYPE solid_angle_measure = REAL;
END_TYPE; -- solid_angle_measure

```

```

TYPE source = ENUMERATION OF
    (made,
     bought,
     not_known);

```

```
END_TYPE; -- source
```

```
TYPE source_item = SELECT
  (identifier);
END_TYPE; -- source_item
```

```
TYPE squared_or_rounded = ENUMERATION OF
  (squared,
   rounded);
END_TYPE; -- squared_or_rounded
```

```
TYPE start_request_item = SELECT
  (product_definition_formation,
   representation_relationship);
END_TYPE; -- start_request_item
```

```
TYPE style_context_select = SELECT
  (group,
   representation,
   representation_item);
END_TYPE; -- style_context_select
```

```
TYPE supported_item = SELECT
  (action_directive,
   action,
   action_method);
END_TYPE; -- supported_item
```

```
TYPE surface_model = SELECT
  (shell_based_surface_model);
END_TYPE; -- surface_model
```

```
TYPE text = STRING;
END_TYPE; -- text
```

```
TYPE text_alignment = label;
END_TYPE; -- text_alignment
```

```
TYPE text_or_character = SELECT
```

```

        (annotation_text,
         annotation_text_character,
         text_literal);
END_TYPE; -- text_or_character

```

```

TYPE text_path = ENUMERATION OF
    (left,
     right,
     up,
     down);
END_TYPE; -- text_path

```

```

TYPE thermodynamic_temperature_measure = REAL;
END_TYPE; -- thermodynamic_temperature_measure

```

```

TYPE time_measure = REAL;
END_TYPE; -- time_measure

```

```

TYPE tolerance_method_definition = SELECT
    (tolerance_value);
END_TYPE; -- tolerance_method_definition

```

```

TYPE tolerance_select = SELECT
    (geometric_tolerance,
     plus_minus_tolerance);
END_TYPE; -- tolerance_select

```

```

TYPE transformation = SELECT
    (item_defined_transformation,
     functionally_defined_transformation);
END_TYPE; -- transformation

```

```

TYPE transition_code = ENUMERATION OF
    (discontinuous,
     continuous,
     cont_same_gradient,
     cont_same_gradient_same_curvature);
END_TYPE; -- transition_code

```

```

TYPE trimming_preference = ENUMERATION OF
    (cartesian,

```

```
        parameter,  
        unspecified);  
END_TYPE; -- trimming_preference
```

```
TYPE trimming_select = SELECT  
    (cartesian_point,  
     parameter_value);  
END_TYPE; -- trimming_select
```

```
TYPE unit = SELECT  
    (named_unit,  
     derived_unit);  
END_TYPE; -- unit
```

```
TYPE value_qualifier = SELECT  
    (precision_qualifier,  
     type_qualifier,  
     uncertainty_qualifier);  
END_TYPE; -- value_qualifier
```

```
TYPE vector_or_direction = SELECT  
    (vector,  
     direction);  
END_TYPE; -- vector_or_direction
```

```
TYPE volume_measure = REAL;  
END_TYPE; -- volume_measure
```

```
TYPE wireframe_model = SELECT  
    (shell_based_wireframe_model,  
     edge_based_wireframe_model);  
END_TYPE; -- wireframe_model
```

```
TYPE work_item = SELECT  
    (product_definition_formation);  
END_TYPE; -- work_item
```

```
TYPE year_number = INTEGER;  
END_TYPE; -- year_number
```

```

ENTITY action;
    name          : label;
    description    : text;
    chosen_method  : action_method;
END_ENTITY; -- action

```

```

ENTITY action_assignment
    ABSTRACT SUPERTYPE;
    assigned_action : action;
END_ENTITY; -- action_assignment

```

```

ENTITY action_directive;
    name          : label;
    description    : text;
    analysis       : text;
    comment        : text;
    requests       : SET [1:?] OF versioned_action_request;
END_ENTITY; -- action_directive

```

```

ENTITY action_method;
    name          : label;
    description    : text;
    consequence    : text;
    purpose        : text;
END_ENTITY; -- action_method

```

```

ENTITY action_method_relationship;
    name          : label;
    description    : text;
    relating_method : action_method;
    related_method  : action_method;
END_ENTITY; -- action_method_relationship

```

```

ENTITY action_relationship;
    name          : label;
    description    : text;
    relating_action : action;
    related_action  : action;
END_ENTITY; -- action_relationship

```

```

ENTITY action_request_assignment
    ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
END_ENTITY; -- action_request_assignment

```



```

ENTITY action_request_solution;
    method      : action_method;
    request      : versioned_action_request;
END_ENTITY; -- action_request_solution

```

```

ENTITY action_request_status;
    status       : label;
    assigned_request : versioned_action_request;
END_ENTITY; -- action_request_status

```

```

ENTITY action_status;
    status       : label;
    assigned_action : executed_action;
END_ENTITY; -- action_status

```

```

ENTITY add_design_object_assignment
    SUBTYPE OF (action_assignment);
    items : SET [1:?] OF managed_design_object;
    WHERE
        WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(it)) AND (it.
            name = 'design object addition')) )) = 1);
END_ENTITY; -- add_design_object_assignment

```

```

ENTITY add_design_object_request_assignment
    SUBTYPE OF (action_request_assignment);
    items : SET [1:?] OF managed_design_object;
    WHERE
        WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(it)) AND (it.
            name = 'design object addition')) )) = 1);
END_ENTITY; -- add_design_object_request_assignment

```

```

ENTITY address;
    internal_location      : OPTIONAL label;
    street_number          : OPTIONAL label;
    street                 : OPTIONAL label;
    postal_box             : OPTIONAL label;
    town                   : OPTIONAL label;
    region                 : OPTIONAL label;
    postal_code            : OPTIONAL label;
    country                : OPTIONAL label;

```

```

    facsimile_number      : OPTIONAL label;
    telephone_number     : OPTIONAL label;
    electronic_mail_address : OPTIONAL label;
    telex_number          : OPTIONAL label;
WHERE
    WR1: ( EXISTS(internal_location) OR EXISTS(street_number) OR EXISTS(
        street) OR EXISTS(postal_box) OR EXISTS(town) OR EXISTS(
        region) OR EXISTS(postal_code) OR EXISTS(country) OR EXISTS(
        facsimile_number) OR EXISTS(telephone_number) OR EXISTS(
        electronic_mail_address) OR EXISTS(telex_number));
END_ENTITY; -- address

```

```

ENTITY advanced_brep_shape_representation
    SUBTYPE OF (shape_representation);
WHERE
    WR1: ( SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MANIFOLD_SOLID_BREP',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FACETED_BREP',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_3D']
        * TYPEOF(it)) = 1)) )) = 0);
    WR2: ( SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MANIFOLD_SOLID_BREP',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM']
        * TYPEOF(it)) = 1) )) > 0);
    WR3: ( SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MANIFOLD_SOLID_BREP'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb_shells(
        msb,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
        | (NOT (SIZEOF(QUERY ( fcs <* csh\
        connected_face_set.cfs_faces | (NOT (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ADVANCED_FACE'
        IN TYPEOF(fcs))) )) = 0)) )) = 0)) )) = 0);
    WR4: ( SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MANIFOLD_SOLID_BREP'
        IN TYPEOF(it)) ) | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ORIENTED_CLOSED_SHELL'
        IN TYPEOF(msb\manifold_solid_brep.outer))) )) = 0);
    WR5: ( SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'BREP_WITH_VOIDS'
            IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* brv\
                brep_with_voids.voids | csh\oriented_closed_shell.
                orientation )) = 0)) )) = 0);
WR6: (SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'
        IN TYPEOF(it)) ) | (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ADVANCED_BREP_SHAPE_REPRESENTATION'
        IN TYPEOF(mi\mapped_item.mapping_source.
        mapped_representation)))) )) = 0);
END_ENTITY; -- advanced_brep_shape_representation

ENTITY advanced_face
    SUBTYPE OF (face_surface);
    WHERE
        WR1 : (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ELEMENTARY_SURFACE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'B_SPLINE_SURFACE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SWEPT_SURFACE']
            * TYPEOF(face_geometry)) = 1);
        WR2 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EDGE_LOOP'
                IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
                elp_fbnds.bound\path.edge_list | (NOT (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EDGE_CURVE'
                IN TYPEOF(oe.edge_element)))) )) = 0)) )) = 0);
        WR3 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EDGE_LOOP'
                IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
                elp_fbnds.bound\path.edge_list | (NOT (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LINE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CONIC',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'POLYLINE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SURFACE_CURVE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'B_SPLINE_CURVE']
                * TYPEOF(oe.edge_element\edge_curve.edge_geometry)) = 1)) ))
                = 0)) )) = 0);
        WR4 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds.bound\path.edge_list | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(oe.edge_start)) AND (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT'
    IN TYPEOF(oe.edge_start\vertex_point.vertex_geometry)) AND
    (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(oe.edge_end)) AND (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT'
    IN TYPEOF(oe.edge_end\vertex_point.vertex_geometry)))))) =
    0)) )) = 0);
WR5 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ORIENTED_PATH'
    IN TYPEOF(elp_fbnds.bound)) )) = 0);
WR6 : ((NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SWEPT_SURFACE'
    IN TYPEOF(face_geometry))) OR (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'B_SPLINE_CURVE']
    * TYPEOF(face_geometry\swept_surface.swept_curve)) = 1));
WR7 : (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(vlp_fbnds\face_bound.bound\vertex_loop.
    loop_vertex)) AND (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT'
    IN TYPEOF(vlp_fbnds\face_bound.bound\vertex_loop.
    loop_vertex\vertex_point.vertex_geometry)))))) = 0);
WR8 : (SIZEOF(QUERY ( bnd <* SELF.bounds | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'EDGE_LOOP',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP']
    * TYPEOF(bnds.bound)) = 1)) )) = 0);
WR9 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds.bound\path.edge_list | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_CURVE'
    IN TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND (
    NOT (SIZEOF(QUERY ( sc_ag <* oe.edge_element\edge_curve.
    edge_geometry\surface_curve.associated_geometry | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PCURVE'
    IN TYPEOF(sc_ag)))) )) = 0))) )) = 0)) )) = 0);
WR10: (((NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SWEPT_SURFACE'
    IN TYPEOF(face_geometry))) OR (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE'
    IN TYPEOF(face_geometry\swept_surface.swept_curve))) OR (
    SIZEOF(face_geometry\swept_surface.swept_curve\polyline.
    points) < 3)) AND (SIZEOF(QUERY ( elp_fbnds <*
    QUERY ( bnds <* SELF.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds.bound\path.edge_list | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE'
    IN TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND (
    NOT (SIZEOF(oe.edge_element\edge_curve.edge_geometry\
    polyline.points) < 3))) )) = 0)) )) = 0));
END_ENTITY; -- advanced_face

```

```

ENTITY alternate_product_relationship;
    name      : label;
    definition : text;
    alternate  : product;
    base       : product;
    basis      : text;
    UNIQUE
        UR1 : alternate, base;
    WHERE
        WR1: (alternate :<>: base);
END_ENTITY; -- alternate_product_relationship

```

```

ENTITY amount_of_substance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AMOUNT_OF_SUBSTANCE_UNIT'
      IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- amount_of_substance_measure_with_unit

ENTITY amount_of_substance_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 1) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- amount_of_substance_unit

ENTITY analytical_model
  SUBTYPE OF (representation);
  UNIQUE
    UR1 : SELF\representation.name;
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 2);
    WR2: (SIZEOF(QUERY ( dr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (dr.assigned_document.
      kind.product_data_type = 'language reference manual') )) = 1);
    WR3: (SIZEOF(QUERY ( it <* SELF.items | (NOT ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MODEL_PARAMETER') IN TYPEOF(it))) )) = 0);
    WR4: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_1') | ((rr.name =
      'access mechanism') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANALYTICAL_MODEL_PORT') IN TYPEOF(rr.rep_2))) )) >= 1);
END_ENTITY; -- analytical_model

ENTITY analytical_model_port
  SUBTYPE OF (representation);
  WHERE

```

```

WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
    name = 'distributed property') AND (it\
    descriptive_representation_item.description IN ['true',
    'false']))) )) = 1);
WR2: (SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
    name = 'nominal signal flow direction') AND (it\
    descriptive_representation_item.description IN [
    'input direction', 'output direction', 'bidirectional',
    'unknown direction', 'not applicable']))) )) <= 1);
WR3: (SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
    name = 'port type') AND (it\descriptive_representation_item.
    description IN ['string property type',
    'logical property type', 'physical property type',
    'boolean property type']))) )) = 1);
WR4: (SIZEOF(QUERY ( am <* QUERY ( rr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') | (rr.name =
    'access mechanism') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANALYTICAL_MODEL') IN TYPEOF(am.rep_1)) )) = 1);
WR5: (SIZEOF(QUERY ( aga <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') | (aga.assigned_group.name
    IN ['scalar port', 'digital scalar port', 'vector port',
    'digital vector port', 'digital analytical model port']) ))
    <= 1);
WR6: ((NOT (SIZEOF(QUERY ( aga <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') | (aga.assigned_group.name
    IN ['vector port', 'digital vector port']) )) = 1)) OR (
    SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
    'size') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COUNT_MEASURE') IN TYPEOF(it\measure_with_unit.
    value_component)) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONTEXT_DEPENDENT_UNIT') IN TYPEOF(it\measure_with_unit.
    unit_component)))) )) = 1));
WR7: ((NOT (SIZEOF(QUERY ( aga <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT.ITEMS') | (aga.assigned_group.name
    IN ['digital scalar port', 'digital vector port',
    'digital analytical model port']) )) = 1)) OR (SIZEOF(
    QUERY ( it <* SELF.items | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'nominal signal flow direction') AND (it\
        descriptive_representation_item.description IN [
        'input direction', 'output direction', 'bidirectional',
        'unknown direction']))) )) = 1));
END_ENTITY; -- analytical_model_port

```

```

ENTITY analytical_representation
  SUBTYPE OF (representation);
  UNIQUE
    UR1 : SELF\representation.name;
  WHERE
    WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PARAMETER_ASSIGNMENT_REPRESENTATION') IN TYPEOF(rr.rep_2)) ))
        >= 1);
    WR2: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL') IN TYPEOF(rr.rep_1)) )) = 1);
END_ENTITY; -- analytical_representation

```

```

ENTITY angular_dimension_with_orientation
  SUBTYPE OF (dimensional_location);
  WHERE
    WR1: (SELF\shape_aspect_relationship.description = 'angular');
END_ENTITY; -- angular_dimension_with_orientation

```

```

ENTITY angular_location
  SUBTYPE OF (dimensional_location);
  angle_selection : angle_relator;
END_ENTITY; -- angular_location

```

```

ENTITY angular_size
  SUBTYPE OF (dimensional_size);
  angle_selection : angle_relator;
END_ENTITY; -- angular_size

```

```

ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
  WHERE

```



```

        WR1: (SELF\geometric_tolerance.name = 'angularity');
END_ENTITY; -- angularity_tolerance

```

```

ENTITY annotation_curve_occurrence
  SUBTYPE OF (annotation_occurrence);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CURVE'
      IN TYPEOF(SELF\styled_item.item));
END_ENTITY; -- annotation_curve_occurrence

```

```

ENTITY annotation_fill_area
  SUBTYPE OF (geometric_representation_item);
  boundaries : SET [1:?] OF curve;
END_ENTITY; -- annotation_fill_area

```

```

ENTITY annotation_fill_area_occurrence
  SUBTYPE OF (annotation_occurrence);
  fill_style_target : point;
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANNOTATION_FILL_AREA'
      IN TYPEOF(SELF.item));
END_ENTITY; -- annotation_fill_area_occurrence

```

```

ENTITY annotation_occurrence
  SUPERTYPE OF (ONEOF (annotation_curve_occurrence,
    annotation_fill_area_occurrence, annotation_text_occurrence,
    annotation_symbol_occurrence))
  SUBTYPE OF (styled_item);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_REPRESENTATION_ITEM'
      IN TYPEOF(SELF));
END_ENTITY; -- annotation_occurrence

```

```

ENTITY annotation_symbol
  SUBTYPE OF (mapped_item);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SYMBOL_REPRESENTATION_MAP'

```

```

        IN TYPEOF(SELF\mapped_item.mapping_source));
    WR2: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SYMBOL_TARGET'
        IN TYPEOF(SELF\mapped_item.mapping_target));
    WR3: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_REPRESENTATION_ITEM'
        IN TYPEOF(SELF));
END_ENTITY; -- annotation_symbol

```

```

ENTITY annotation_symbol_occurrence
    SUBTYPE OF (annotation_occurrence);
    WHERE
        WR1: (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ANNOTATION_SYMBOL',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DEFINED_SYMBOL']
            * TYPEOF(SELF\styled_item.item)) > 0);
END_ENTITY; -- annotation_symbol_occurrence

```

```

ENTITY annotation_text
    SUBTYPE OF (mapped_item);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'AXIS2_PLACEMENT'
            IN TYPEOF(SELF\mapped_item.mapping_target));
        WR2: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TEXT_STRING_REPRESENTATION'
            IN TYPEOF(SELF\mapped_item.mapping_source.
                mapped_representation));
        WR3: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_REPRESENTATION_ITEM'
            IN TYPEOF(SELF));
END_ENTITY; -- annotation_text

```

```

ENTITY annotation_text_character
    SUBTYPE OF (mapped_item);
    alignment : text_alignment;
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CHARACTER_GLYPH_SYMBOL'
            IN TYPEOF(SELF\mapped_item.mapping_source.

```

```

        mapped_representation));
WR2: (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT'
    IN TYPEOF(SELF\mapped_item.mapping_target));
WR3: (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_REPRESENTATION_ITEM'
    IN TYPEOF(SELF));
END_ENTITY; -- annotation_text_character

ENTITY annotation_text_occurrence
    SUBTYPE OF (annotation_occurrence);
    WHERE
        WR1: (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TEXT_LITERAL',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ANNOTATION_TEXT',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ANNOTATION_TEXT_CHARACTER',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DEFINED_CHARACTER_GLYPH',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPOSITE_TEXT']
            * TYPEOF(SELF\styled_item.item)) > 0);
END_ENTITY; -- annotation_text_occurrence

ENTITY apex
    SUBTYPE OF (derived_shape_aspect);
END_ENTITY; -- apex

ENTITY application_context;
    application : text;
    INVERSE
        context_elements : SET [1:?] OF application_context_element FOR
            frame_of_reference;
END_ENTITY; -- application_context

ENTITY application_context_element
    SUPERTYPE OF (ONEOF (product_context,product_definition_context,
        product_concept_context));
    name : label;
    frame_of_reference : application_context;
END_ENTITY; -- application_context_element

```

```

ENTITY application_protocol_definition;
    status : label;
    application_interpreted_model_schema_name : label;
    application_protocol_year : year_number;
    application : application_context;
END_ENTITY; -- application_protocol_definition

```

```

ENTITY applied_action_assignment
    SUBTYPE OF (action_assignment);
    items : SET [1:?] OF action_assigned_item;
END_ENTITY; -- applied_action_assignment

```

```

ENTITY applied_approval_assignment
    SUBTYPE OF (approval_assignment);
    items : SET [1:?] OF approval_assigned_item;
END_ENTITY; -- applied_approval_assignment

```

```

ENTITY applied_certification_assignment
    SUBTYPE OF (certification_assignment);
    items : SET [1:?] OF certification_assigned_item;
END_ENTITY; -- applied_certification_assignment

```

```

ENTITY applied_classification_assignment
    SUBTYPE OF (group_assignment);
    items : SET [1:?] OF classification_assigned_item;
END_ENTITY; -- applied_classification_assignment

```

```

ENTITY applied_contract_assignment
    SUBTYPE OF (contract_assignment);
    items : SET [1:?] OF contract_assigned_item;
END_ENTITY; -- applied_contract_assignment

```

```

ENTITY applied_date_and_time_assignment
    SUBTYPE OF (date_and_time_assignment);
    items : SET [1:?] OF date_and_time_assigned_item;
    WHERE
        WR1: applied_date_time_correlation(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
;
        WR2: ((NOT (SELF.role.name = 'participant date and time')) OR (
            SIZEOF(QUERY ( ra <* QUERY ( it <* SELF.items | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'RULE_ACTION') IN TYPEOF(it)) ) | (NOT (SIZEOF(USEDIN(ra,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')) = 1)) ))
        = 0));
END_ENTITY; -- applied_date_and_time_assignment

```

```

ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_assigned_item;
  WHERE
    WR1: applied_date_correlation(SELF,

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
;
END_ENTITY; -- applied_date_assignment

```

```

ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_assigned_item;
END_ENTITY; -- applied_document_reference

```

```

ENTITY applied_group_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF group_assigned_item;
END_ENTITY; -- applied_group_assignment

```

```

ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_assigned_item;
END_ENTITY; -- applied_organization_assignment

```

```

ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_assigned_item;
END_ENTITY; -- applied_person_and_organization_assignment

```

```

ENTITY applied_person_assignment
  SUBTYPE OF (person_assignment);
  items : SET [1:?] OF person_assigned_item;
END_ENTITY; -- applied_person_assignment

```

```

ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_assigned_item;
END_ENTITY; -- applied_security_classification_assignment

```

```

ENTITY approval;
    status : approval_status;
    level : label;
END_ENTITY; -- approval

```

```

ENTITY approval_assignment
    ABSTRACT SUPERTYPE;
    assigned_approval : approval;
END_ENTITY; -- approval_assignment

```

```

ENTITY approval_date_time;
    date_time : date_time_select;
    dated_approval : approval;
END_ENTITY; -- approval_date_time

```

```

ENTITY approval_person_organization;
    person_organization : person_organization_select;
    authorized_approval : approval;
    role : approval_role;
END_ENTITY; -- approval_person_organization

```

```

ENTITY approval_relationship;
    name : label;
    description : text;
    relating_approval : approval;
    related_approval : approval;
END_ENTITY; -- approval_relationship

```

```

ENTITY approval_role;
    role : label;
END_ENTITY; -- approval_role

```

```

ENTITY approval_status;
    name : label;
END_ENTITY; -- approval_status

```

```

ENTITY area_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'AREA_UNIT'
        IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- area_measure_with_unit

ENTITY area_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 2) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- area_unit

ENTITY array_placement_group_component_definition
  SUBTYPE OF (assembly_group_component_definition);
END_ENTITY; -- array_placement_group_component_definition

ENTITY array_placement_group_component_shape_aspect
  SUBTYPE OF (assembly_group_component_shape_aspect);
END_ENTITY; -- array_placement_group_component_shape_aspect

ENTITY assembly_bond_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MATERIAL_DESIGNATION.DEFINITIONS')) = 1);
    WR2: (SIZEOF(QUERY ( aga <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT.ITEMS') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BOND_CATEGORY') IN TYPEOF(aga.assigned_group)) )) = 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'bonded feature 1') )) = 1);
    WR4: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'bonded feature 2') )) = 1);
END_ENTITY; -- assembly_bond_definition

```

```

ENTITY assembly_component_usage
  SUPERTYPE OF (ONEOF (next_assembly_usage_occurrence,
    specified_higher_usage_occurrence, promissory_usage_occurrence))
  SUBTYPE OF (product_definition_usage);
  reference_designator : OPTIONAL identifier;
END_ENTITY; -- assembly_component_usage

```

```

ENTITY assembly_component_usage_substitute;
  name      : label;
  definition : text;
  base      : assembly_component_usage;
  substitute : assembly_component_usage;
  UNIQUE
    UR1 : base, substitute;
  WHERE
    WR1: (base.relatering_product_definition :=: substitute.
      relatering_product_definition);
    WR2: (base :<>: substitute);
END_ENTITY; -- assembly_component_usage_substitute

```

```

ENTITY assembly_definition
  SUBTYPE OF (physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'assembly module') )) >= 1);
    WR2: ((NOT (SELF.frame_of_reference.name = 'physical design')) OR (
      SIZEOF(QUERY ( du <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
      | ((SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_DEFINITION',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_ASSEMBLY_DEFINITION'] * TYPEOF(du)) = 1)
      AND (du.related_product_definition.frame_of_reference.name =
      'physical design usage') AND (SIZEOF(QUERY ( prpc <*
      USEDIN(du.related_product_definition.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
      product_category.name = 'assembly module') )) >= 1)) ) = 1));
    WR3: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```



```

        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(sa)) AND (sa.
        description = 'pca terminal')) )) >= 2)) )) = 0));
END_ENTITY; -- assembly_definition

```

```

ENTITY assembly_group_component_definition
  SUBTYPE OF (component_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( gc <* QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
    | (pdr.name = 'group component') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION') IN TYPEOF(gc.
    related_product_definition)) )) >= 1);
END_ENTITY; -- assembly_group_component_definition

```

```

ENTITY assembly_group_component_shape_aspect
  SUBTYPE OF (component_shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( gc <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'group component') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT') IN TYPEOF(gc.related_shape_aspect)) ))
    >= 1);
END_ENTITY; -- assembly_group_component_shape_aspect

```

```

ENTITY assembly_joint
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
  WHERE
    WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL') IN TYPEOF(SELF.relateing_shape_aspect))
    AND (SELF.relateing_shape_aspect.description IN [
    'assembly module component terminal',
    'bare die component terminal',
    'interconnect component join terminal',
    'interconnect module component terminal',
    'package terminal occurrence',
    'packaged component join terminal'])) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_INTERFACE_TERMINAL') IN TYPEOF(SELF.
    relateing_shape_aspect)));
    WR2: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'COMPONENT_TERMINAL') IN TYPEOF(SELF.related_shape_aspect))
AND (SELF.related_shape_aspect.description IN [
    'assembly module component terminal',
    'bare die component terminal',
    'interconnect component join terminal',
    'interconnect module component terminal',
    'package terminal occurrence',
    'packaged component join terminal']) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_INTERFACE_TERMINAL') IN TYPEOF(SELF.
related_shape_aspect));
WR3: ((NOT (SELF\shape_aspect_relationship.name =
    'assembled with bonding')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
    SELF.of_shape.definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
    | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(pdr)) )) >= 1));
WR4: ((NOT (SELF\shape_aspect_relationship.name =
    'assembled with bonding')) OR (SIZEOF(QUERY ( ddu <*
    QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'default definition usage') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_BOND_DEFINITION') IN TYPEOF(ddu.
    relating_shape_aspect)) )) = 1));
WR5: ((NOT (SELF\shape_aspect_relationship.name =
    'assembled with bonding')) OR (SIZEOF(QUERY ( pd <* USEDIN(
    SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
    used_representation.name = 'bond assembly position') AND (
    SIZEOF(QUERY ( it <* pdr.used_representation.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM') IN TYPEOF(it)) AND (it.name =
    'assembly position')) )) = 1)) )) = 1) )) <= 1));
WR6: ((NOT (SELF\shape_aspect_relationship.name =
    'assembled with fasteners')) OR (SIZEOF(QUERY ( pdr <*
    USEDIN(SELF.of_shape.definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
    | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(pdr)) )) >= 1));
END_ENTITY; -- assembly_joint

```

```

ENTITY assembly_module_terminal
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition\
      product_definition.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'assembly module') )) >= 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
      >= 1)) )) = 0);
    WR3: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'member connected terminal') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_CONNECTED_TERMINALS_DEFINITION') IN TYPEOF(mct.
      relating_shape_aspect)) )) <= 1);
    WR4: ((NOT (SELF\shape_aspect.description = 'pca terminal')) OR ((
      SIZEOF(TYPEOF(SELF.of_shape.definition) * [
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_DEFINITION',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_ASSEMBLY_DEFINITION']) >= 1) AND (SELF.
      of_shape.definition\product_definition.frame_of_reference.
      name = 'physical design usage')));
    WR5: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      related_shape_aspect.description = 'connection zone') )) = 1);
    WR6: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      related_shape_aspect.description = 'interface terminal') ))
      <= 1);
  END_ENTITY; -- assembly_module_terminal

```

```

ENTITY assembly_module_usage_view_connector_relationship
  SUBTYPE OF (mapped_item);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_3D') IN TYPEOF(SELF\mapped_item.
      mapping_target));

```

```

WR2: (SELF\mapped_item.mapping_source.mapped_representation.name =
      '3d bound volume shape');
WR3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF\mapped_item.mapping_source.
      mapped_representation,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
      (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_CONNECTOR',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PACKAGED_CONNECTOR',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PACKAGED_CONNECTOR'] * TYPEOF(pdr.
      definition.definition)) = 1) )) = 1);
WR4: (SIZEOF(QUERY ( r <* USEDIN(SELF\representation_item,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (SIZEOF(QUERY ( pdr <* USEDIN(r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
      ((pdr.definition.definition.frame_of_reference\
      application_context_element.name = 'physical design usage')
      AND (SIZEOF(QUERY ( pc <* USEDIN(pdr.definition.definition.
      of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc\
      product_category.name = 'assembly module') )) = 1)) )) = 1) ))
      = 1);
WR5: (SIZEOF(QUERY ( sr <* USEDIN(SELF\representation_item,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (sr\representation.
      context_of_items\geometric_representation_context.
      coordinate_space_dimension = 3) )) = SIZEOF(QUERY ( sr <*
      USEDIN(SELF\representation_item,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (sr <> []) )));
END_ENTITY; -- assembly_module_usage_view_connector_relationship

```

```

ENTITY auxiliary_characteristic_dimension_representation
  SUBTYPE OF (dimensional_characteristic_representation);
END_ENTITY; -- auxiliary_characteristic_dimension_representation

```

```

ENTITY axis1_placement
  SUBTYPE OF (placement);
  axis : OPTIONAL direction;
  DERIVE
    z : direction := NVL(normalise(axis),dummy_gri || direction([0,0,1]));
  WHERE
    WR1: (SELF\geometric_representation_item.dim = 3);
END_ENTITY; -- axis1_placement

```

```

ENTITY axis2_placement_2d
  SUBTYPE OF (placement);
    ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [2:2] OF direction := build_2axes(ref_direction);
  WHERE
    WR1: (SELF\geometric_representation_item.dim = 2);
END_ENTITY; -- axis2_placement_2d

ENTITY axis2_placement_3d
  SUBTYPE OF (placement);
    axis          : OPTIONAL direction;
    ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [3:3] OF direction := build_axes(axis,ref_direction);
  WHERE
    WR1: (SELF\placement.location.dim = 3);
    WR2: ((NOT EXISTS(axis)) OR (axis.dim = 3));
    WR3: ((NOT EXISTS(ref_direction)) OR (ref_direction.dim = 3));
    WR4: ((NOT EXISTS(axis)) OR (NOT EXISTS(ref_direction)) OR (
      cross_product(axis,ref_direction).magnitude > 0));
END_ENTITY; -- axis2_placement_3d

ENTITY b_spline_curve
  SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
    quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve)
  SUBTYPE OF (bounded_curve);
    degree          : INTEGER;
    control_points_list : LIST [2:?] OF cartesian_point;
    curve_form       : b_spline_curve_form;
    closed_curve      : LOGICAL;
    self_intersect     : LOGICAL;
  DERIVE
    upper_index_on_control_points : INTEGER := SIZEOF(
      control_points_list) - 1;
    control_points                 : ARRAY [0:
      upper_index_on_control_points] OF
      cartesian_point := list_to_array(
        control_points_list,0,
        upper_index_on_control_points);
  WHERE
    WR1: ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'UNIFORM_CURVE'
      IN TYPEOF(SELF)) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'QUASI_UNIFORM_CURVE'
      IN TYPEOF(SELF)) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'BEZIER_CURVE'
        IN TYPEOF(SELF)) OR (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'B_SPLINE_CURVE_WITH_KNOTS'
        IN TYPEOF(SELF)));
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
  knot_multiplicities : LIST [2:?] OF INTEGER;
  knots               : LIST [2:?] OF parameter_value;
  knot_spec           : knot_type;
  DERIVE
    upper_index_on_knots : INTEGER := SIZEOF(knots);
  WHERE
    WR1: constraints_param_b_spline(degree, upper_index_on_knots,
      upper_index_on_control_points, knot_multiplicities, knots);
    WR2: (SIZEOF(knot_multiplicities) = upper_index_on_knots);
END_ENTITY; -- b_spline_curve_with_knots

ENTITY b_spline_surface
  SUPERTYPE OF (ONEOF (b_spline_surface_with_knots, uniform_surface,
    quasi_uniform_surface, bezier_surface) ANDOR
    rational_b_spline_surface)
  SUBTYPE OF (bounded_surface);
  u_degree      : INTEGER;
  v_degree      : INTEGER;
  control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
  surface_form   : b_spline_surface_form;
  u_closed       : LOGICAL;
  v_closed       : LOGICAL;
  self_intersect : LOGICAL;
  DERIVE
    u_upper      : INTEGER := SIZEOF(control_points_list) - 1;
    v_upper      : INTEGER := SIZEOF(control_points_list[1]) - 1;
    control_points : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
      cartesian_point := make_array_of_array(
        control_points_list, 0, u_upper, 0, v_upper);
  WHERE
    WR1: ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'UNIFORM_SURFACE'
      IN TYPEOF(SELF)) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'QUASI_UNIFORM_SURFACE'
      IN TYPEOF(SELF)) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BEZIER_SURFACE'
      IN TYPEOF(SELF)) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'B_SPLINE_SURFACE_WITH_KNOTS'
        IN TYPEOF(SELF)));
END_ENTITY; -- b_spline_surface

```

```

ENTITY b_spline_surface_with_knots
  SUBTYPE OF (b_spline_surface);
    u_multiplicities : LIST [2:?] OF INTEGER;
    v_multiplicities : LIST [2:?] OF INTEGER;
    u_knots          : LIST [2:?] OF parameter_value;
    v_knots          : LIST [2:?] OF parameter_value;
    knot_spec        : knot_type;
  DERIVE
    knot_u_upper : INTEGER := SIZEOF(u_knots);
    knot_v_upper : INTEGER := SIZEOF(v_knots);
  WHERE
    WR1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,
      knot_u_upper, SELF\b_spline_surface.u_upper, u_multiplicities,
      u_knots);
    WR2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
      knot_v_upper, SELF\b_spline_surface.v_upper, v_multiplicities,
      v_knots);
    WR3: (SIZEOF(u_multiplicities) = knot_u_upper);
    WR4: (SIZEOF(v_multiplicities) = knot_v_upper);
END_ENTITY; -- b_spline_surface_with_knots

```

```

ENTITY bare_die
  SUBTYPE OF (physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( ifdu <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATED_PRODUCT_DEFINITION') | (pdr.name =
      'implemented function') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT') IN TYPEOF(ifdu.
      relating_product_definition)) AND (ifdu.
      relating_product_definition.frame_of_reference.name =
      'functional design usage')) ) = 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( dut <*
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | (pdr.name = 'device unit technology') ) | (dut.
      relating_property_definition.name = 'unit technology') )) =
      1) )) = 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```

```

        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0);
END_ENTITY; -- bare_die

```

```

ENTITY bare_die_terminal
  SUBTYPE OF (minimally_defined_bare_die_terminal);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE') IN TYPEOF(SELF.of_shape.definition));
    WR2: (SIZEOF(QUERY ( eca <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'external connection area') ) | (eca.
    related_shape_aspect.description = 'connection zone') )) = 1);
    WR3: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL DESIGNATION.DEFINITIONS')) = 1);
END_ENTITY; -- bare_die_terminal

```

```

ENTITY bezier_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve

```

```

ENTITY bezier_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

```

```

ENTITY block
  SUBTYPE OF (geometric_representation_item);
  position : axis2_placement_3d;
  x         : positive_length_measure;
  y         : positive_length_measure;
  z         : positive_length_measure;
END_ENTITY; -- block

```

```

ENTITY bond_category
  SUBTYPE OF (group, externally_defined_item);
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
END_ENTITY; -- bond_category

```



```

ENTITY boolean_result
  SUBTYPE OF (geometric_representation_item);
    operator      : boolean_operator;
    first_operand  : boolean_operand;
    second_operand : boolean_operand;
END_ENTITY; -- boolean_result

```

```

ENTITY boundary_curve
  SUBTYPE OF (composite_curve_on_surface);
  WHERE
    WR1: SELF\composite_curve.closed_curve;
END_ENTITY; -- boundary_curve

```

```

ENTITY bounded_curve
  SUPERTYPE OF (ONEOF (polyline,b_spline_curve,trimmed_curve,
    bounded_pcurve,bounded_surface_curve,composite_curve))
  SUBTYPE OF (curve);
END_ENTITY; -- bounded_curve

```

```

ENTITY bounded_pcurve
  SUBTYPE OF (pcurve, bounded_curve);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BOUNDED_CURVE'
      IN TYPEOF(SELF\pcurve.reference_to_curve.items[1]));
END_ENTITY; -- bounded_pcurve

```

```

ENTITY bounded_surface
  SUPERTYPE OF (ONEOF (b_spline_surface,rectangular_trimmed_surface,
    curve_bounded_surface,rectangular_composite_surface))
  SUBTYPE OF (surface);
END_ENTITY; -- bounded_surface

```

```

ENTITY bounded_surface_curve
  SUBTYPE OF (surface_curve, bounded_curve);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BOUNDED_CURVE'
      IN TYPEOF(SELF\surface_curve.curve_3d));
END_ENTITY; -- bounded_surface_curve

```

```

ENTITY box_domain;
  corner : cartesian_point;
  xlength : positive_length_measure;
  ylength : positive_length_measure;
  zlength : positive_length_measure;
WHERE
  WR1: (SIZEOF(QUERY ( item <* USEDIN(SELF, '' ) | (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOXED_HALF_SPACE'
    IN TYPEOF(item))) )) = 0);
END_ENTITY; -- box_domain

ENTITY boxed_half_space
  SUBTYPE OF (half_space_solid);
  enclosure : box_domain;
END_ENTITY; -- boxed_half_space

ENTITY brep_with_voids
  SUBTYPE OF (manifold_solid_brep);
  voids : SET [1:?] OF oriented_closed_shell;
END_ENTITY; -- brep_with_voids

ENTITY bus_structural_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1: (SIZEOF(QUERY ( bce <* QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
    | (pdr.name = 'bus composition element') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'NETWORK_NODE_DEFINITION') IN TYPEOF(bce.
    related_product_definition)) AND (bce.
    related_product_definition.frame_of_reference.name =
    'functional network design')) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(bce.
    related_product_definition))) )) >= 1);
  WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
    | (pdr.name =
    'functional unit network terminal definition bus assignment') ))
    <= 1) )) <= 1);
END_ENTITY; -- bus_structural_definition

```

```

ENTITY calendar_date
  SUBTYPE OF (date);
    day_component : day_in_month_number;
    month_component : month_in_year_number;
  WHERE
    WR1: valid_calendar_date(SELF);
END_ENTITY; -- calendar_date

```

```

ENTITY cartesian_point
  SUBTYPE OF (point);
    coordinates : LIST [1:3] OF length_measure;
END_ENTITY; -- cartesian_point

```

```

ENTITY cartesian_transformation_operator
  SUPERTYPE OF (ONEOF (cartesian_transformation_operator_2d,
    cartesian_transformation_operator_3d))
  SUBTYPE OF (geometric_representation_item,
    functionally_defined_transformation);
    axis1 : OPTIONAL direction;
    axis2 : OPTIONAL direction;
    local_origin : cartesian_point;
    scale : OPTIONAL REAL;
  DERIVE
    scl : REAL := NVL(scale,1);
  WHERE
    WR1: (scl > 0);
END_ENTITY; -- cartesian_transformation_operator

```

```

ENTITY cartesian_transformation_operator_2d
  SUBTYPE OF (cartesian_transformation_operator);
  DERIVE
    u : LIST [2:2] OF direction := base_axis(2,SELF\
      cartesian_transformation_operator.axis1,SELF\
      cartesian_transformation_operator.axis2,?);
  WHERE
    WR1: (SELF\geometric_representation_item.dim = 2);
END_ENTITY; -- cartesian_transformation_operator_2d

```

```

ENTITY cartesian_transformation_operator_3d
  SUBTYPE OF (cartesian_transformation_operator);
    axis3 : OPTIONAL direction;
  DERIVE
    u : LIST [3:3] OF direction := base_axis(3,SELF\
      cartesian_transformation_operator.axis1,SELF\
      cartesian_transformation_operator.axis2,axis3);
  WHERE

```

```

        WR1: (SELF\geometric_representation_item.dim = 3);
END_ENTITY; -- cartesian_transformation_operator_3d

```

```

ENTITY centre_of_symmetry
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( sadr <* deriving_relationships | (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SYMMETRIC_SHAPE_ASPECT'
      IN TYPEOF(sadr.related_shape_aspect)))) ) = 0);
END_ENTITY; -- centre_of_symmetry

```

```

ENTITY certification;
  name      : label;
  purpose   : text;
  kind      : certification_type;
END_ENTITY; -- certification

```

```

ENTITY certification_assignment
  ABSTRACT SUPERTYPE;
  assigned_certification : certification;
END_ENTITY; -- certification_assignment

```

```

ENTITY certification_type;
  description : label;
END_ENTITY; -- certification_type

```

```

ENTITY change
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF work_item;
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DIRECTED_ACTION') IN TYPEOF(SELF.assigned_action));
    WR2: unique_version_change_order(SELF.assigned_action,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
;
END_ENTITY; -- change

```

```

ENTITY change_from_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
  WHERE
    WR1: (SIZEOF(QUERY ( aa <* USEDIN(SELF\action_assignment.

```

```

        assigned_action,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CHANGE_TO_DESIGN_OBJECT_ASSIGNMENT') IN TYPEOF(aa)) )) >= 1);
END_ENTITY; -- change_from_design_object_assignment

```

```

ENTITY change_from_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
  WHERE
    WR1: (SIZEOF(QUERY ( aa <* USEDIN(SELF\action_request_assignment.
      assigned_action_request,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CHANGE_TO_DESIGN_OBJECT_REQUEST_ASSIGNMENT') IN TYPEOF(aa)) ))
      >= 1);
END_ENTITY; -- change_from_design_object_request_assignment

```

```

ENTITY change_request
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF change_request_item;
END_ENTITY; -- change_request

```

```

ENTITY change_to_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
  WHERE
    WR1: (SIZEOF(QUERY ( aa <* USEDIN(SELF\action_assignment.
      assigned_action,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CHANGE_FROM_DESIGN_OBJECT_ASSIGNMENT') IN TYPEOF(aa)) )) >=
      1);
    WR2: (SIZEOF(QUERY ( pdr <* SELF.items | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(pdr)) AND (pdr.
      name = 'design object change')) )) = 1);
END_ENTITY; -- change_to_design_object_assignment

```

```

ENTITY change_to_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
  WHERE
    WR1: (SIZEOF(QUERY ( aa <* USEDIN(SELF\action_request_assignment.

```

```

        assigned_action_request,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CHANGE_FROM_DESIGN_OBJECT_REQUEST_ASSIGNMENT') IN TYPEOF(aa)) ))
    >= 1);
    WR2: (SIZEOF(QUERY ( pdr <* SELF.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(pdr)) AND (pdr.
        name = 'design object change')) )) = 1);
END_ENTITY; -- change_to_design_object_request_assignment

```

```

ENTITY character_glyph_font_usage;
    character : character_glyph_symbol;
    font      : text_font;
END_ENTITY; -- character_glyph_font_usage

```

```

ENTITY character_glyph_style_outline;
    outline_style : curve_style;
END_ENTITY; -- character_glyph_style_outline

```

```

ENTITY character_glyph_style_stroke;
    stroke_style : curve_style;
END_ENTITY; -- character_glyph_style_stroke

```

```

ENTITY character_glyph_symbol
    SUBTYPE OF (symbol_representation);
    character_box : planar_extent;
    baseline_ratio : ratio_measure;
    DERIVE
        box_height : length_measure := character_box.size_in_y;
    WHERE
        WR1: ((0 <= baseline_ratio) AND (baseline_ratio <= 1));
        WR2: item_in_context(SELF.character_box, SELF\representation.
            context_of_items);
        WR3: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'POSITIVE_LENGTH_MEASURE'
            IN TYPEOF(SELF.box_height));
END_ENTITY; -- character_glyph_symbol

```

```

ENTITY character_glyph_symbol_outline
    SUBTYPE OF (character_glyph_symbol);
    outlines : SET [1:?] OF annotation_fill_area;
    WHERE

```

```

        WR1: (SELF.outlines <= SELF\representation.items);
END_ENTITY; -- character_glyph_symbol_outline

```

```

ENTITY characterized_object;
    name      : label;
    description : text;
END_ENTITY; -- characterized_object

```

```

ENTITY characterized_product_category
    SUBTYPE OF (characterized_object, product_category);
    UNIQUE
        UR1 : SELF\product_category.name;
    WHERE
        WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | ((pd.description =
            'product category values') AND (SIZEOF(QUERY ( pdr <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PARAMETER_ASSIGNMENT_REPRESENTATION') IN TYPEOF(pdr.
            used_representation)) )) >= 1)) )) = 1);
        WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | ((pd.description =
            'product category parameters') AND (SIZEOF(QUERY ( pdr <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
            QUERY ( ri <* pdr.used_representation.items | (NOT (SIZEOF(
            TYPEOF(ri) * [
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MODEL_PARAMETER']) = 1)) )) = 0) )) >= 1)) )) = 1);
        WR3: (SELF\characterized_object.name = SELF\product_category.name);
END_ENTITY; -- characterized_product_category

```

```

ENTITY circle
    SUBTYPE OF (conic);
    radius : positive_length_measure;
END_ENTITY; -- circle

```

```

ENTITY circular_runout_tolerance
    SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
    WHERE
        WR1: (SELF\geometric_tolerance.name = 'circular runout');
        WR2: (NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'

```

```

        + 'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF)));
    WR3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.name = 'referenced datum system definition') AND (
        pdr.related_property_definition.description =
        'datum system property without material conditions')) )) = 1);
END_ENTITY; -- circular_runout_tolerance

```

```

ENTITY closed_shell
    SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell

```

```

ENTITY colour;
END_ENTITY; -- colour

```

```

ENTITY colour_rgb
    SUBTYPE OF (colour_specification);
    red    : REAL;
    green  : REAL;
    blue   : REAL;
    WHERE
        WR1: ((0 <= red) AND (red <= 1));
        WR2: ((0 <= green) AND (green <= 1));
        WR3: ((0 <= blue) AND (blue <= 1));
END_ENTITY; -- colour_rgb

```

```

ENTITY colour_specification
    SUBTYPE OF (colour);
    name : label;
END_ENTITY; -- colour_specification

```

```

ENTITY component_definition
    SUPERTYPE OF (ONEOF (printed_component, packaged_component) ANDOR
        thermal_component)
    SUBTYPE OF (product_definition);
    WHERE
        WR1 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_DEFINITION') | (NOT (SIZEOF(
            QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION_DEFINITION') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )

```



```

| (sr_pdr.used_representation.name =
'planar projected shape')) <= 1)) = 0);
WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation))) )
| (sr_pdr.used_representation.name =
'3d bound volume shape')) <= 1)) = 0);
WR3 : ((NOT (SELF\product_definition.description =
'bare die component')) OR (SIZEOF(QUERY ( ip <*
QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| (pdr.name = 'instantiated part')) | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'BARE_DIE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_BARE_DIE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_BARE_DIE'] * TYPEOF(ip.
relating_product_definition)) = 1) AND (ip.
relating_product_definition.frame_of_reference.name =
'physical design usage')) = 1));
WR4 : ((NOT ((SELF.frame_of_reference.name = 'physical occurrence')
AND (SIZEOF(QUERY ( pc1 <* USEDIN(SELF.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc1\
product_category.name = 'assembly module')) >= 1))) OR (
SIZEOF(QUERY ( ip <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| (pdr.name = 'instantiated part')) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT') IN TYPEOF(ip.relying_product_definition))
AND (ip.relying_product_definition.frame_of_reference.name
= 'physical design usage') AND (SIZEOF(QUERY ( pc2 <*
USEDIN(ip.relying_product_definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc2\
product_category.name = 'assembly module')) >= 1))) = 1));
WR5 : ((NOT ((SELF.frame_of_reference.name = 'physical occurrence')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'assembly module')) >= 1))) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL') IN TYPEOF(sa)) AND (sa.description =
'assembly module component terminal')) )) >= 2) )) >= 1));
WR6 : ((NOT ((SELF.frame_of_reference.name = 'physical occurrence')
AND (SIZEOF(QUERY ( pc1 <* USEDIN(SELF.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc1\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( ip <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| (pdr.name = 'instantiated part') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT') IN TYPEOF(ip.relying_product_definition))
AND (ip.relying_product_definition.frame_of_reference.name
= 'physical design usage') AND (SIZEOF(QUERY ( pc2 <*
USEDIN(ip.relying_product_definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc2\
product_category.name = 'interconnect module') )) >= 1)) ))
= 1));
WR7 : ((NOT ((SELF.frame_of_reference.name = 'physical occurrence')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL') IN TYPEOF(sa)) AND (sa.description =
'interconnect module component terminal')) )) >= 2) )) >= 1));
WR8 : ((NOT (SELF\product_definition.description =
'mating connector')) OR (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS') IN TYPEOF(
SELF)) AND (SIZEOF(QUERY ( doc <* SELF\
product_definition_with_associated_documents.
documentation_ids | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EE_SPECIFICATION') IN TYPEOF(doc)) )) = 1)) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```

```

QUERY ( pt_occ <* QUERY ( sa <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (sa.description =
'part template occurrence') ) | (SIZEOF(QUERY ( it <*
QUERY ( sar <* USEDIN(pt_occ,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relying_shape_aspect.
description = 'printed connector template')) )) = 1) )) = 1)) ))
= 0) OR (SIZEOF(QUERY ( ip <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| (pdr.name = 'instantiated part') ) | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNALLY_DEFINED_PACKAGED_PART',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(ip.
relating_product_definition)) = 1) AND (ip.
relating_product_definition.description =
'packaged connector')) )) = 1));
WR9 : ((NOT (SELF\product_definition.description =
'mating connector')) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name = 'mating connector placement') ))
= 1) )) = 1));
WR10: ((NOT (SELF\product_definition.description =
'mating connector')) OR ((SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
| ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'NEXT_ASSEMBLY_USAGE_OCCURRENCE') IN TYPEOF(pdr)) )) = 0)
AND (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'NEXT_ASSEMBLY_USAGE_OCCURRENCE') IN TYPEOF(pdr)) )) = 0)))));
WR11: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
| ((pdr.name = 'instantiated part') AND (NOT (SELF.
formation :=: pdr.relying_product_definition.formation)))) ))

```

```

        = 0);
WR12: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )
        | ((sr_pdr.used_representation.name =
        'part template non planar 2d shape') OR (sr_pdr.
        used_representation.name = 'non planar 2d shape') OR (
        sr_pdr.used_representation.name =
        'open shell based surface')) )) <= 1)) )) = 0);
END_ENTITY; -- component_definition

```

```

ENTITY component_functional_terminal
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_FUNCTIONAL_UNIT') IN TYPEOF(SELF.of_shape.
        definition));
    WR2: (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated terminal') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT_TERMINAL_DEFINITION') IN TYPEOF(it.
        relating_shape_aspect)) )) = 1);
    WR3: (SIZEOF(QUERY ( futba <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'functional unit terminal bus assignment') )
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(futba.
        relating_property_definition.definition)) )) <= 1);
    WR4: (SIZEOF(QUERY ( futna <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'functional unit terminal node assignment') )
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NETWORK_NODE_DEFINITION') IN TYPEOF(futna.
        relating_property_definition.definition)) )) <= 1);
END_ENTITY; -- component_functional_terminal

```

```

ENTITY component_functional_unit
  SUBTYPE OF (product_definition);
  WHERE

```

```

WR1: (SELF.frame_of_reference.name = 'functional occurrence');
WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_FUNCTIONAL_TERMINAL') IN TYPEOF(sa)) )) >= 1) ))
    >= 1);
WR3: (SIZEOF(QUERY ( ifu <* QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | (pdr.name = 'instantiated functional unit') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT') IN TYPEOF(ifu.relying_product_definition)) ))
    = 1);
WR4: (SIZEOF(QUERY ( nc <* QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | (pdr.name = 'network composition') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT') IN TYPEOF(nc.relying_product_definition))
    AND (nc.relying_product_definition.frame_of_reference.name
    = 'functional network design')) )) = 1);
END_ENTITY; -- component_functional_unit

```

ENTITY component_interface_terminal

SUBTYPE OF (shape_aspect);

WHERE

```

WR1 : (SELF\shape_aspect.description IN [
    'interconnect component interface terminal',
    'packaged connector component interface terminal']);
WR2 : (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition));
WR3 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'instantiated feature') )) <= 1);
WR4 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'design usage') )) <= 1);
WR5 : ((NOT (SELF\shape_aspect.description =
    'packaged connector component interface terminal')) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERFACE_COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.
    definition)));
WR6 : ((NOT (SELF\shape_aspect.description =
    'packaged connector component interface terminal')) OR (
    SELF.of_shape.definition\product_definition.

```

```

        frame_of_reference.name = 'physical occurrence'));
WR7 : ((NOT (SELF\shape_aspect.description =
        'interconnect component interface terminal')) OR ((SIZEOF(
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        related_shape_aspect.description = 'connection zone')) =
        1) AND (NOT (SELF.of_shape.definition\product_definition.
        description IN ['assembly module component',
        'bare die component', 'interconnect module component',
        'laminant component', 'packaged component'])))));
WR8 : ((NOT (SELF\shape_aspect.description =
        'packaged connector component interface terminal')) OR SELF
        .product_definitional);
WR9 : ((NOT (SELF\shape_aspect.description =
        'packaged connector component interface terminal')) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_COMPONENT') IN TYPEOF(SELF.of_shape.definition))
        AND (SELF.of_shape.definition\product_definition.
        description = 'packaged connector component') AND (SELF.
        of_shape.definition\product_definition.frame_of_reference.
        name = 'physical occurrence')));
WR10: ((NOT (SELF\shape_aspect.description =
        'packaged connector component interface terminal')) OR (
        SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature')) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART_TERMINAL') IN TYPEOF(i_f.
        relating_shape_aspect)) AND (i_f.relying_shape_aspect.
        description = 'interface terminal')) )) = 1));
END_ENTITY; -- component_interface_terminal

```

ENTITY component_location

SUBTYPE OF (representation);

WHERE

```

WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'placement fixed') AND ((it\
        descriptive_representation_item.description = 'true') OR (it
        \descriptive_representation_item.description = 'false')))) ))
        = 1);
WR2: (SIZEOF(QUERY ( ms <* QUERY ( it <* SELF.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'mounting style')) ) | (NOT (ms\
        descriptive_representation_item.description IN ['normal',
        'reversed']))) )) = 0);
WR3: ((NOT (SELF.context_of_items\geometric_representation_context.

```

```

coordinate_space_dimension = 2)) OR (SIZEOF(QUERY ( it <*
SELF.items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_TRANSFORMATION_OPERATOR_2D') IN TYPEOF(it)) )) =
1));
WR4: ((NOT (SELF.context_of_items\geometric_representation_context.
coordinate_space_dimension = 3)) OR (SIZEOF(QUERY ( it <*
SELF.items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_3D') IN TYPEOF(it)) )) = 1));
END_ENTITY; -- component_location

ENTITY component_shape_aspect
  SUPERTYPE OF (ONEOF (land,inter_stratum_feature) ANDOR
    thermal_component_shape_aspect)
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1 : (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition));
    WR2 : ((SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTER_STRATUM_FEATURE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROBE_ACCESS_AREA',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LAND'] * TYPEOF(SELF)) = 1) OR (SELF\shape_aspect.
      description IN ['component feature',
        'component termination passage interface terminal',
        'land interface terminal',
        'non functional land interface terminal',
        'printed connector component interface terminal',
        'lamine component','lamine text component',
        'lamine text string component',
        'conductive interconnect element with pre defined transitions',
        'conductive interconnect element with user defined single transition',
        'routed interconnect component',
        'special symbol lamine component',
        'primary stratum indicator symbol',
        'stratum feature template component',
        'interconnect module component surface feature']));
    WR3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )
      | (sr_pdr.used_representation.name =
        'planar projected shape') )) <= 1)) )) = 0);
    WR4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )
| (sr_pdr.used_representation.name =
'3d bound volume shape') )) <= 1)) )) = 0);
WR5 : ((NOT (SELF\shape_aspect.description IN ['component feature',
'component termination passage interface terminal',
'land interface terminal',
'non functional land interface terminal',
'printed connector component interface terminal']))) OR (
SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'design usage') )) <= 1));
WR6 : ((NOT (SELF\shape_aspect.description IN [
'laminate text component', 'laminate text string component',
'special symbol laminate component',
'primary stratum indicator symbol',
'stratum feature template component']))) OR (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition
\product_definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))));
WR7 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') )) <= 1);
WR8 : ((NOT (SELF.description =
'component termination passage interface terminal')) OR (
SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE') IN TYPEOF(ac.relating_shape_aspect)) AND
(ac.relating_shape_aspect.description =
'component termination passage')) )) = 1));
WR9 : ((NOT (SELF\shape_aspect.description =
'component termination passage interface terminal')) OR (
SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') ) | (i_f.
relating_shape_aspect.description =
'component termination passage template interface terminal') ))

```



```

= 1));
WR10: ((NOT (SELF\shape_aspect.description =
'laminate text component')) OR (SIZEOF(QUERY ( it <*
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEXT_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect))) = 1));
WR11: ((NOT (SELF\shape_aspect.description IN [
'stratum feature template component']))) OR (SIZEOF(
QUERY ( sfi <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'stratum feature implementation') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE') IN TYPEOF(sfi.relying_shape_aspect))) =
1));
WR12: ((NOT (SELF\shape_aspect.description IN [
'land interface terminal',
'non functional land interface terminal']))) OR (SIZEOF(
QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_TEMPLATE_TERMINAL') IN TYPEOF(i_f.
relating_shape_aspect)) AND (i_f.related_shape_aspect.
description = 'interface terminal')) = 1));
WR13: ((NOT (SELF\shape_aspect.description =
'land interface terminal')) OR (SIZEOF(QUERY ( ac <*
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND') IN TYPEOF(ac.relying_shape_aspect)) AND (ac.
related_shape_aspect.description IN ['functional land',
'via dependent land', 'via and contact size dependent land',
'component termination passage dependent land',
'contact size dependent land',
'component termination passage and contact size dependent land']))) =
1));
WR14: ((NOT (SELF\shape_aspect.description =
'non functional land interface terminal')) OR (SIZEOF(
QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND') IN TYPEOF(ac.relying_shape_aspect)) AND (ac.

```

```

        related_shape_aspect.description IN ['non functional land',
        'via dependent non functional land',
        'via and contact size dependent non functional land',
        'component termination passage dependent non functional land',
        'contact size dependent non functional land',
        'component termination passage and contact size ' +
        'dependent non functional land',
        'unsupported passage dependent non functional land']))) )) =
        1));
WR15: ((NOT (SELF\shape_aspect.description IN [
        'conductive interconnect element with pre defined transitions',
        'conductive interconnect element with ' +
        'user defined single transition']))) OR (SIZEOF(
        QUERY ( cc <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'composed conductor') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(cc.related_shape_aspect))) )) =
        1));
WR16: ((NOT (SELF\shape_aspect.description IN [
        'conductive interconnect element with pre defined transitions',
        'conductive interconnect element with user ' +
        'defined single transition']))) OR (SIZEOF(QUERY ( ac <*
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL') IN TYPEOF(ac.related_shape_aspect))
        AND (ac.related_shape_aspect.description =
        'conductive interconnect element terminal')) )) = 2));
WR17: ((NOT (SELF\shape_aspect.description =
        'printed connector component interface terminal')) OR SELF.
        product_definitional);
WR18: ((NOT (SELF\shape_aspect.description =
        'printed connector component interface terminal')) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRINTED_COMPONENT') IN TYPEOF(SELF.of_shape.definition))
        AND (SELF.of_shape.definition\product_definition.
        description = 'printed connector component')));
WR19: ((NOT (SELF\shape_aspect.description =
        'printed connector component interface terminal')) OR (
        SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(i_f.
        relating_shape_aspect)) AND (i_f.relying_shape_aspect.
        description = 'interface terminal')) )) = 1));
WR20: ((NOT (SELF\shape_aspect.description =

```

```

        'routed interconnect component')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) >= 1) ))
= 1));
WR21: ((NOT (SELF\shape_aspect.description IN [
'special symbol laminate component',
'primary stratum indicator symbol'])) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.related_shape_aspect.
description = 'special symbol part template')) )) = 1));
WR22: ((NOT (SELF\shape_aspect.description =
'stratum feature template component')) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relateing_shape_aspect.
description = 'stratum feature template')) OR ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
relating_shape_aspect))) )) = 1));
WR23: ((NOT (SELF\shape_aspect.description IN [
'laminate text component'])) OR (SIZEOF(QUERY ( sfi <*
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'stratum feature implementation') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE') IN TYPEOF(sfi.relateing_shape_aspect)) ))
>= 1));
WR24: ((NOT (SELF\shape_aspect.description =
'interconnect module component surface feature')) OR (
SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') ) | (i_f.
relating_shape_aspect.description IN [
'interconnect module secondary surface',
'interconnect module primary surface',
'interconnect module edge surface',
'interconnect module edge segment surface']) )) = 1));
END_ENTITY; -- component_shape_aspect

```

```

ENTITY component_terminal
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1 : (SELF\shape_aspect.description IN [
      'assembly module component terminal',
      'bare die component terminal',
      'component termination passage join terminal',
      'conductive interconnect element terminal',
      'interconnect component join terminal',
      'interconnect module component terminal',
      'land join terminal', 'minimally defined component terminal',
      'non functional land join terminal',
      'packaged component join terminal',
      'printed component join terminal',
      'package terminal occurrence', 'via terminal']);
    WR2 : (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition));
    WR3 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'instantiated feature') )) <= 1);
    WR4 : (SIZEOF(QUERY ( at <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'associated terminals') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(at.
      relating_shape_aspect)) )) <= 1);
    WR5 : (SIZEOF(QUERY ( cr <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'connectivity requirement') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_NETWORK') IN TYPEOF(cr.relatating_shape_aspect)) ))
      <= 1);
    WR6 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'design usage') )) <= 1);
    WR7 : ((NOT (SELF\shape_aspect.description IN [
      'assembly module component terminal',
      'bare die component terminal', 'package terminal occurrence',
      'packaged component join terminal',
      'printed component join terminal'])) OR SELF.
      product_definitional);
    WR8 : ((NOT (SELF\shape_aspect.description =
      'assembly module component terminal')) OR ((SELF.of_shape.
      definition.frame_of_reference.name = 'physical occurrence')
      AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition
      \product_definition.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1));
WR9 : ((NOT (SELF\shape_aspect.description =
        'assembly module component terminal')) OR (SIZEOF(
        QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(i_f.
        relating_shape_aspect))) = 1));
WR10: ((NOT (SELF\shape_aspect.description =
        'bare die component terminal')) OR (SELF.of_shape.
        definition\product_definition.description =
        'bare die component')));
WR11: ((NOT (SELF\shape_aspect.description =
        'bare die component terminal')) OR (SIZEOF(QUERY ( i_f <*
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_TERMINAL') IN TYPEOF(i_f.relying_shape_aspect))) =
        1));
WR12: ((NOT (SELF\shape_aspect.description =
        'component termination passage join terminal')) OR (SIZEOF(
        QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_PASSAGE') IN TYPEOF(ac.relying_shape_aspect)) AND
        (ac.relying_shape_aspect.description =
        'component termination passage')))) = 1));
WR13: ((NOT (SELF\shape_aspect.description =
        'component termination passage join terminal')) OR (SIZEOF(
        QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | (i_f.
        relating_shape_aspect.description =
        'component termination passage template join terminal') ))
        = 1));
WR14: ((NOT (SELF\shape_aspect.description =
        'conductive interconnect element terminal')) OR (SIZEOF(
        QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(ac.
        relating_shape_aspect)) AND ((ac.relying_shape_aspect.

```

```

        description =
        'conductive interconnect element with pre defined transitions')
        OR (ac.relating_shape_aspect.description =
        'conductive interconnect element with user defined single transition')))) ))
        = 1));
WR15: ((NOT (SELF\shape_aspect.description IN [
        'conductive interconnect element terminal',
        'interconnect component join terminal',
        'printed component join terminal'])) OR (SIZEOF(
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        related_shape_aspect.name = 'connection zone') )) = 1));
WR16: ((NOT (SELF\shape_aspect.description =
        'interconnect module component terminal')) OR ((SELF.
        of_shape.definition\product_definition.frame_of_reference.
        name = 'physical occurrence') AND (SIZEOF(QUERY ( prpc <*
        USEDIN(SELF.of_shape.definition\product_definition.
        formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
        product_category.name = 'interconnect module') )) >= 1))));
WR17: ((NOT (SELF\shape_aspect.description =
        'interconnect module component terminal')) OR (SIZEOF(
        QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_TERMINAL') IN TYPEOF(i_f.
        relating_shape_aspect)) AND (SIZEOF(QUERY ( pc <* USEDIN(
        i_f.relating_shape_aspect.of_shape.definition\
        product_definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc\
        product_category.name = 'interconnect module') )) > 0)) ))
        = 1));
WR18: ((NOT (SELF\shape_aspect.description IN ['land join terminal',
        'non functional land join terminal'])) OR (SIZEOF(
        QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_TEMPLATE_TERMINAL') IN TYPEOF(i_f.
        relating_shape_aspect)) AND (i_f.related_shape_aspect.
        description = 'join terminal')) )) = 1));
WR19: ((NOT (SELF\shape_aspect.description = 'land join terminal'))
        OR (SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND') IN TYPEOF(ac.relate_shape_aspect)) AND (ac.
        relate_shape_aspect.description IN ['functional land',
        'via dependent land', 'via and contact size dependent land',
        'component termination passage dependent land',
        'contact size dependent land',
        'component termination passage and contact size dependent land']))) ))
        = 1));
WR20: ((NOT (SELF\shape_aspect.description =
        'minimally defined terminal')) OR (SIZEOF(QUERY ( pdr <*
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'part terminal external reference') )) = 1));
WR21: ((NOT (SELF\shape_aspect.description =
        'non functional land join terminal')) OR (SIZEOF(
        QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND') IN TYPEOF(ac.relate_shape_aspect)) AND (ac.
        relate_shape_aspect.description IN ['non functional land',
        'via dependent non functional land',
        'via and contact size dependent non functional land',
        'component termination passage dependent non functional land',
        'contact size dependent non functional land',
        'component termination passage and contact size ' +
        'dependent non functional land',
        'unsupported passage dependent non functional land']))) )) =
        1));
WR22: ((NOT (SELF\shape_aspect.description =
        'package terminal occurrence')) OR (SIZEOF(QUERY ( ud <*
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'usage definition') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_TERMINAL') IN TYPEOF(ud.relate_shape_aspect)) ))
        = 1));
WR23: ((NOT (SELF\shape_aspect.description IN [
        'package terminal occurrence',
        'packaged component join terminal']))) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_COMPONENT') IN TYPEOF(SELF.of_shape.definition))
        AND (SELF.of_shape.definition\product_definition.
        frame_of_reference.name = 'physical occurrence')));
WR24: ((NOT (SELF\shape_aspect.description =
        'packaged component join terminal')) OR (SIZEOF(
        QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_PART_TERMINAL') IN TYPEOF(i_f.
relating_shape_aspect)) AND (i_f.relating_shape_aspect.
description = 'join terminal')) )) = 1));
WR25: ((NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) )) <= 2) ))
<= 1) )) <= 1));
WR26: ((NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( lmwu <* QUERY ( it <* pdr.used_representation.items
| (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) ) | (lmwu.
name = 'maximum wire length') )) <= 1) )) <= 1) )) <= 1));
WR27: ((NOT (SELF\shape_aspect.description =
'packaged component join terminal')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( lmwu <* QUERY ( it <* pdr.used_representation.items
| (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) ) | (lmwu.
name = 'minimum wire length') )) <= 1) )) <= 1) )) <= 1));
WR28: ((NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR ((SELF.of_shape.

```



```

definition\product_definition.frame_of_reference.name =
'physical occurrence') AND (SIZEOF(QUERY ( prpc <* USEDIN(
SELF.of_shape.definition\product_definition.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)));
WR29: ((NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR (SIZEOF(
QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated feature') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(i_f.
relating_shape_aspect)) AND (i_f.relateing_shape_aspect.
description = 'join terminal')) )) = 1));
WR30: ((NOT (SELF\shape_aspect.description =
'printed component join terminal')) OR (SIZEOF(
QUERY ( i <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (pdr.
name = 'implementation') ) | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT'] * TYPEOF(i.relateing_shape_aspect))
>= 1) OR (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION') IN TYPEOF(i.relateing_shape_aspect.
of_shape.definition)) AND (i.relateing_shape_aspect.of_shape
.definition\product_definition.frame_of_reference.name =
'physical occurrence') AND (SIZEOF(QUERY ( pc <* USEDIN(i.
relateing_shape_aspect.of_shape.definition\
product_definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (pc\
product_category.name = 'interconnect module') )) >= 1))) ))
= 1));
WR31: ((NOT (SELF\shape_aspect.description = 'via terminal')) OR (
SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE') IN TYPEOF(ac.relateing_shape_aspect)) AND
(ac.relateing_shape_aspect.description IN ['buried via',
'interfacial connection', 'bonded conductive base blind via',
'non conductive base blind via',
'plated conductive base blind via']))) )) = 1));

```

```

WR32: ((NOT (SELF\shape_aspect.description = 'via terminal')) OR (
    SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'instantiated feature') ) | (i_f.
        relating_shape_aspect.description = 'via template terminal') ))
    = 1));
WR33: ((NOT (SELF\shape_aspect.description =
    'interconnect component join terminal')) OR ((SELF.of_shape
    .definition\product_definition.frame_of_reference.name =
    'physical occurrence') AND (NOT (SELF.of_shape.definition\
    product_definition.description IN [
    'assembly module component', 'bare die component',
    'interconnect module component', 'lamine component',
    'packaged component'])))));
WR34: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( dri <* pdr.used_representation.items | ((dri.name =
    'global swappable') AND (dri.description IN ['true',
    'false']))) )) = 1) )) = 1) )) <= 1);
WR35: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( dri <* pdr.used_representation.items | ((dri.name =
    'local swappable') AND (dri.description IN ['true', 'false']))) ))
    = 1) )) = 1) )) <= 1);
WR36: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( dri <* pdr.used_representation.items | (dri.name =
    'swap code') )) = 1) )) = 1) )) <= 1);
END_ENTITY; -- component_terminal

ENTITY composite_array_shape_aspect
    SUBTYPE OF (composite_shape_aspect);
END_ENTITY; -- composite_array_shape_aspect

ENTITY composite_curve
    SUBTYPE OF (bounded_curve);
    segments : LIST [1:?] OF composite_curve_segment;

```

```

        self_intersect : LOGICAL;
    DERIVE
        n_segments      : INTEGER := SIZEOF(segments);
        closed_curve    : LOGICAL := segments[n_segments].transition <>
                                discontinuous;
    WHERE
        WR1: (((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
            temp.transition = discontinuous) )) = 1)) OR (closed_curve
            AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
            discontinuous) )) = 0)));
    END_ENTITY; -- composite_curve

```

```

ENTITY composite_curve_on_surface
    SUPERTYPE OF (boundary_curve)
    SUBTYPE OF (composite_curve);
    DERIVE
        basis_surface : SET [0:2] OF surface := get_basis_surface(SELF);
    WHERE
        WR1: (SIZEOF(basis_surface) > 0);
        WR2: constraints_composite_curve_on_surface(SELF);
    END_ENTITY; -- composite_curve_on_surface

```

```

ENTITY composite_curve_segment
    SUBTYPE OF (founded_item);
        transition      : transition_code;
        same_sense      : BOOLEAN;
        parent_curve    : curve;
    INVERSE
        using_curves : BAG [1:?] OF composite_curve FOR segments;
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'BOUNDED_CURVE'
            IN TYPEOF(parent_curve));
    END_ENTITY; -- composite_curve_segment

```

```

ENTITY composite_group_shape_aspect
    SUBTYPE OF (composite_shape_aspect);
    END_ENTITY; -- composite_group_shape_aspect

```

```

ENTITY composite_shape_aspect
    SUBTYPE OF (shape_aspect);
    INVERSE
        component_relationships : SET [2:?] OF shape_aspect_relationship FOR
            relating_shape_aspect;
    END_ENTITY; -- composite_shape_aspect

```

```

ENTITY composite_unit_shape_aspect
  SUBTYPE OF (composite_shape_aspect);
END_ENTITY; -- composite_unit_shape_aspect

```

```

ENTITY concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
  WHERE
    WR1: (SELF\geometric_tolerance.name = 'concentricity');
END_ENTITY; -- concentricity_tolerance

```

```

ENTITY configuration_design;
  configuration : configuration_item;
  design       : product_definition_formation;
  UNIQUE
    UR1 : configuration, design;
END_ENTITY; -- configuration_design

```

```

ENTITY configuration_effectivity
  SUBTYPE OF (product_definition_effectivity);
  configuration : configuration_design;
  UNIQUE
    UR1 : configuration, SELF\product_definition_effectivity.usage,
        SELF\effectivity.id;
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_USAGE'
      IN TYPEOF(SELF\product_definition_effectivity.usage));
END_ENTITY; -- configuration_effectivity

```

```

ENTITY configuration_item;
  id       : identifier;
  name     : label;
  description : OPTIONAL text;
  item_concept : product_concept;
  purpose  : OPTIONAL label;
  UNIQUE
    UR1 : id;
END_ENTITY; -- configuration_item

```

```

ENTITY conic
  SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
  SUBTYPE OF (curve);
  position : axis2_placement;
END_ENTITY; -- conic

```

```

ENTITY conical_surface
  SUBTYPE OF (elementary_surface);
    radius      : length_measure;
    semi_angle  : plane_angle_measure;
  WHERE
    WR1: (radius >= 0);
END_ENTITY; -- conical_surface


ENTITY connected_edge_set
  SUBTYPE OF (topological_representation_item);
    ces_edges : SET [1:?] OF edge;
END_ENTITY; -- connected_edge_set


ENTITY connected_face_set
  SUPERTYPE OF (ONEOF (closed_shell, open_shell))
  SUBTYPE OF (topological_representation_item);
    cfs_faces : SET [1:?] OF face;
END_ENTITY; -- connected_face_set


ENTITY connection_zone_based_assembly_joint
  SUBTYPE OF (assembly_joint);
  WHERE
    WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'zone_1') )) = 1);
    WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'zone_2') )) = 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      relating_shape_aspect.description = 'connection zone') )) =
      2);
END_ENTITY; -- connection_zone_based_assembly_joint


ENTITY connection_zone_interface_plane_relationship
  SUBTYPE OF (shape_aspect_relationship);
  UNIQUE
    UR1 : SELF\shape_aspect_relationship.name;
  WHERE
    WR1: (SIZEOF(TYPEOF(SELF)) = 2);
    WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SEATING_PLANE') IN TYPEOF(SELF\shape_aspect_relationship.
      related_shape_aspect));

```

```

        WR3: (SELF\shape_aspect_relationship.related_shape_aspect.
              description = 'plane');
        WR4: (SELF\shape_aspect_relationship.relateing_shape_aspect.
              description = 'connection zone');
END_ENTITY; -- connection_zone_interface_plane_relationship

ENTITY connectivity_sub_structure
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL') IN TYPEOF(sar.related_shape_aspect))
      OR (sar.related_shape_aspect.description =
        'topological junction')) )) > 1);
END_ENTITY; -- connectivity_sub_structure

ENTITY context_dependent_shape_representation;
  representation_relation      : shape_representation_relationship;
  represented_product_relation : product_definition_shape;
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP'
      IN TYPEOF(SELF.represented_product_relation.definition));
END_ENTITY; -- context_dependent_shape_representation

ENTITY context_dependent_unit
  SUBTYPE OF (named_unit);
  name : label;
END_ENTITY; -- context_dependent_unit

ENTITY contract;
  name      : label;
  purpose   : text;
  kind      : contract_type;
END_ENTITY; -- contract

ENTITY contract_assignment
  ABSTRACT SUPERTYPE;
  assigned_contract : contract;
END_ENTITY; -- contract_assignment

```

```

ENTITY contract_type;
    description : label;
END_ENTITY; -- contract_type

```

```

ENTITY conversion_based_unit
    SUBTYPE OF (named_unit);
    name : label;
    conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit

```

```

ENTITY coordinated_representation_item
    SUBTYPE OF (representation, representation_item);
    WHERE
        WR1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') ) +
            (SIZEOF(USEDIN(pdr,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DATA_ENVIRONMENT.ELEMENTS')) <= 1) )) <= 1);
        WR2: (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
        WR3: ((NOT (SELF\representation_item.name = 'tolerance')) OR (
            SIZEOF(QUERY ( it <* SELF.items | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) )) >= 1));
        WR4: ((NOT (SELF\representation_item.name = 'plus minus tolerance'))
            OR (SIZEOF(QUERY ( it <* SELF.items | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) )) = 3));
        WR5: ((NOT (SELF\representation_item.name = 'symmetrical tolerance'))
            OR (SIZEOF(QUERY ( it <* SELF.items | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) )) = 2));
        WR6: ((NOT (SELF\representation_item.name = 'plus minus tolerance'))
            OR ((SIZEOF(QUERY ( it <* SELF.items | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
            'basic value')) )) = 1) AND (SIZEOF(QUERY ( it <* SELF.
            items | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
            'plus value')) )) = 1) AND (SIZEOF(QUERY ( it <* SELF.items
            | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
            'minus value')) )) = 1)))));
        WR7: ((NOT (SELF\representation_item.name = 'symmetrical tolerance'))
            OR ((SIZEOF(QUERY ( it <* SELF.items | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
        'basic value')) )) = 1) AND (SIZEOF(QUERY ( it <* SELF.
items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name =
'deviation value')) )) = 1)));
END_ENTITY; -- coordinated_representation_item

```

```

ENTITY coordinated_universal_time_offset;
    hour_offset      : hour_in_day;
    minute_offset    : OPTIONAL minute_in_hour;
    sense            : ahead_or_behind;
END_ENTITY; -- coordinated_universal_time_offset

```

```

ENTITY csg_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
        WR1: (SELF.context_of_items\geometric_representation_context.
            coordinate_space_dimension = 3);
        WR2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CSG_SOLID',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'AXIS2_PLACEMENT_3D',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SOLID_REPLICA',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REVOLVED_FACE_SOLID',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EXTRUDED_FACE_SOLID']
            * TYPEOF(it)) <> 1) )) = 0);
        WR3: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM'
            IN TYPEOF(it)) ) | (NOT (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CSG_SHAPE_REPRESENTATION'
            IN TYPEOF(mi\mapped_item.mapping_source.
            mapped_representation))) )) = 0);
        WR4: (SIZEOF(QUERY ( sr <* QUERY ( it <* SELF.items | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SOLID_REPLICA'
            IN TYPEOF(it)) ) | (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CSG_SOLID',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REVOLVED_FACE_SOLID',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

        'EXTRUDED_FACE_SOLID']
        * TYPEOF(sr\solid_replica.parent_solid)) = 0) )) = 0);
WR5: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D'
        IN TYPEOF(it))) )) > 0);
END_ENTITY; -- csg_shape_representation

ENTITY csg_solid
    SUBTYPE OF (solid_model);
    tree_root_expression : csg_select;
END_ENTITY; -- csg_solid

ENTITY curve
    SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve,offset_curve_2d,
        offset_curve_3d,curve_replica))
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- curve

ENTITY curve_bounded_surface
    SUBTYPE OF (bounded_surface);
    basis_surface : surface;
    boundaries : SET [1:?] OF boundary_curve;
    implicit_outer : BOOLEAN;
    WHERE
        WR1: (NOT (implicit_outer AND (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'OUTER_BOUNDARY_CURVE'
                IN TYPEOF(boundaries))));
        WR2: ((NOT implicit_outer) OR (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'BOUNDED_SURFACE'
                IN TYPEOF(basis_surface)));
        WR3: (SIZEOF(QUERY ( temp <* boundaries | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'OUTER_BOUNDARY_CURVE'
                IN TYPEOF(temp)) )) <= 1);
        WR4: (SIZEOF(QUERY ( temp <* boundaries | (temp\
            composite_curve_on_surface.basis_surface[1] <> SELF.
            basis_surface) )) = 0);
END_ENTITY; -- curve_bounded_surface

ENTITY curve_dimension
    SUBTYPE OF (dimensional_size);
    WHERE
        WR1: (SELF\dimensional_size.name <> 'angular');
END_ENTITY; -- curve_dimension

```

```

ENTITY curve_replica
  SUBTYPE OF (curve);
    parent_curve    : curve;
    transformation   : cartesian_transformation_operator;
  WHERE
    WR1: (transformation.dim = parent_curve.dim);
    WR2: acyclic_curve_replica(SELF, parent_curve);
END_ENTITY; -- curve_replica

```

```

ENTITY curve_style;
  name          : label;
  curve_font     : curve_font_or_scaled_curve_font_select;
  curve_width    : size_select;
  curve_colour   : colour;
END_ENTITY; -- curve_style

```

```

ENTITY curve_style_font;
  name          : label;
  pattern_list  : LIST [1:?] OF curve_style_font_pattern;
END_ENTITY; -- curve_style_font

```

```

ENTITY curve_style_font_and_scaling;
  name          : label;
  curve_font     : curve_style_font_select;
  curve_font_scaling : REAL;
END_ENTITY; -- curve_style_font_and_scaling

```

```

ENTITY curve_style_font_pattern;
  visible_segment_length : positive_length_measure;
  invisible_segment_length : positive_length_measure;
END_ENTITY; -- curve_style_font_pattern

```

```

ENTITY curve_style_with_ends_and_corners
  SUBTYPE OF (curve_style);
    curve_ends      : squared_or_rounded;
    curve_corners   : squared_or_rounded;
END_ENTITY; -- curve_style_with_ends_and_corners

```

```

ENTITY curve_style_with_extension
  SUBTYPE OF (curve_style);
    curve_extensions : length_measure;
END_ENTITY; -- curve_style_with_extension

```

```

ENTITY cutout_edge_segment
  SUPERTYPE OF (plated_cutout_edge_segment)
  SUBTYPE OF (inter_stratum_feature, shape_aspect_relationship);
  WHERE
    WR1: (SELF.description IN ['plated cutout edge segment',
      'cutout edge segment']);
    WR2: (SIZEOF(QUERY ( cc <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'composed cutout') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTER_STRATUM_FEATURE') IN TYPEOF(cc.relate_shape_aspect))
      AND (cc.relate_shape_aspect.description IN ['cutout',
      'physical connectivity interrupting cutout','plated cutout']))))
      = 1);
    WR3: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
      shape_aspect_relationship.relate_shape_aspect));
    WR4: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect));
    WR5: (SELF\shape_aspect_relationship.related_shape_aspect :<=: SELF\
      shape_aspect_relationship.relate_shape_aspect);
END_ENTITY; -- cutout_edge_segment

```

```

ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

```

```

ENTITY data_environment;
  name      : label;
  description : text;
  elements  : SET [1:?] OF property_definition_representation;
END_ENTITY; -- data_environment

```

```

ENTITY date
  SUPERTYPE OF (calendar_date);
  year_component : year_number;
END_ENTITY; -- date

```

```

ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY; -- date_and_time

```

```

ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
    assigned_date_and_time : date_and_time;
    role                    : date_time_role;
END_ENTITY; -- date_and_time_assignment

```

```

ENTITY date_assignment
  ABSTRACT SUPERTYPE;
    assigned_date : date;
    role          : date_role;
END_ENTITY; -- date_assignment

```

```

ENTITY date_role;
  name : label;
END_ENTITY; -- date_role

```

```

ENTITY date_time_role;
  name : label;
END_ENTITY; -- date_time_role

```

```

ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
    effectivity_start_date : date_and_time;
    effectivity_end_date   : OPTIONAL date_and_time;
END_ENTITY; -- dated_effectivity

```

```

ENTITY datum_difference
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  WHERE
    WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PHYSICAL_UNIT_DATUM') IN TYPEOF(SELF\
              shape_aspect_relationship.relate_shape_aspect)) AND ((
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PHYSICAL_UNIT_DATUM') IN TYPEOF(SELF\
              shape_aspect_relationship.related_shape_aspect))));
    WR2: (SELF\shape_aspect.name = SELF\shape_aspect_relationship.name);
    WR3: (SELF\shape_aspect.description = SELF\shape_aspect_relationship
          .description);
    WR4: (SELF\shape_aspect_relationship.relate_shape_aspect :<>: SELF
          \shape_aspect_relationship.related_shape_aspect);
    WR5: (SIZEOF(TYPEOF(SELF)) = 3);
END_ENTITY; -- datum_difference

```

```

ENTITY datum_difference_analytical_model_port_assignment
  SUBTYPE OF (property_definition_representation);
  WHERE
    WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANALYTICAL_MODEL_PORT') IN TYPEOF(SELF\
      property_definition_representation.used_representation)) AND
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
      + 'DATUM_DIFFERENCE') IN TYPEOF(SELF\
      property_definition_representation.definition)));
    WR5: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- datum_difference_analytical_model_port_assignment

```

```

ENTITY datum_difference_based_characteristic
  SUBTYPE OF (representation_item);
  WHERE
    WR1: (SIZEOF(QUERY ( r <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (SIZEOF(QUERY ( prd <* USEDIN(r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
      (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
      + 'PROPERTY_DEFINITION_REPRESENTATION') IN TYPEOF(prd)) AND
      (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE') IN TYPEOF(prd\
      property_definition_representation.definition)))) )) = 1) ))
      = 1);
END_ENTITY; -- datum_difference_based_characteristic

```

```

ENTITY datum_difference_based_model_parameter
  SUBTYPE OF (model_parameter);
  WHERE
    WR1: (SIZEOF(QUERY ( r <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (NOT (SIZEOF(QUERY ( pdr <*
      USEDIN(r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION') IN TYPEOF(pdr)) AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE') IN TYPEOF(pdr\
      property_definition_representation.definition)))) )) = 0)) ))
      = 1);
END_ENTITY; -- datum_difference_based_model_parameter

```

```

ENTITY datum_difference_functional_unit_usage_view_terminal_assignment
  SUBTYPE OF (shape_aspect_relationship);
  WHERE

```

```

WR1: (SIZEOF(TYPEOF(SELF)) = 2);
WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT_TERMINAL_DEFINITION') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect));
WR3: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_DIFFERENCE') IN TYPEOF(SELF\shape_aspect_relationship
      .relating_shape_aspect));
END_ENTITY; -- datum_difference_functional_unit_usage_view_terminal_assignment

```

ENTITY datum_reference_frame

SUBTYPE OF (shape_aspect);

WHERE

```

WR1: (SIZEOF(TYPEOF(SELF)) = 2);
WR2: (SELF\shape_aspect.product_definitional = FALSE);
WR3: (SIZEOF(QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'reference axis') ) | ((pud.related_shape_aspect.
      description = 'axis') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) ) ) <= 3);
WR4: (SIZEOF(QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'reference plane') ) | ((pud.related_shape_aspect.
      description = 'plane') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) ) ) <= 3);
WR5: (SIZEOF(QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'reference origin') ) | ((pud.related_shape_aspect.
      description = 'point') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) ) ) <= 1);
WR6: (SIZEOF(QUERY ( ds <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'established datum reference frame') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DATUM_SYSTEM') IN TYPEOF(ds)) ) ) <= 1);
WR7: ((SIZEOF(QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'reference axis') ) | ((pud.related_shape_aspect.
      description = 'axis') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) ) ) + SIZEOF(
      QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.

```

```

        name = 'reference plane') ) | ((pud.related_shape_aspect.
        description = 'plane') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) )) + SIZEOF(
        QUERY ( pud <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'reference origin') ) | ((pud.related_shape_aspect.
        description = 'point') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT_DATUM') IN TYPEOF(pud))) ))) >= 1);
END_ENTITY; -- datum_reference_frame

ENTITY datum_system
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SELF\shape_aspect.product_definitional = FALSE);
    WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'established datum reference frame') )) = 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'datum usage in datum system') )) >= 1);
    WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((pd.description =
        'datum system property with material conditions') OR (pd.
        description =
        'datum system property without material conditions')) )) = 1);
END_ENTITY; -- datum_system

ENTITY datum_system_based_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY; -- datum_system_based_dimensional_location

ENTITY definitional_representation
  SUBTYPE OF (representation);
  WHERE
    WR1: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PARAMETRIC_REPRESENTATION_CONTEXT'
        IN TYPEOF(SELF\representation.context_of_items));
END_ENTITY; -- definitional_representation

ENTITY degenerate_pcurve

```

```

SUBTYPE OF (point);
  basis_surface      : surface;
  reference_to_curve : definitional_representation;
WHERE
  WR1: (SIZEOF(reference_to_curve\representation.items) = 1);
  WR2: (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE'
      IN TYPEOF(reference_to_curve\representation.items[1]));
  WR3: (reference_to_curve\representation.items[1]\
    geometric_representation_item.dim = 2);
END_ENTITY; -- degenerate_pcurve

ENTITY degenerate_toroidal_surface
  SUBTYPE OF (toroidal_surface);
  select_outer : BOOLEAN;
WHERE
  WR1: (major_radius < minor_radius);
END_ENTITY; -- degenerate_toroidal_surface

ENTITY delete_design_object_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE
  WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(it)) AND (it.
    name = 'design object deletion')) )) = 1);
END_ENTITY; -- delete_design_object_assignment

ENTITY delete_design_object_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF managed_design_object;
WHERE
  WR1: (SIZEOF(QUERY ( it <* SELF.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(it)) AND (it.
    name = 'design object deletion')) )) = 1);
END_ENTITY; -- delete_design_object_request_assignment

ENTITY derived_shape_aspect
  SUPERTYPE OF (ONEOF (apex,centre_of_symmetry,geometric_alignment,
    geometric_intersection,extension,tangent))
  SUBTYPE OF (shape_aspect);
  INVERSE
    deriving_relationships : SET [1:?] OF shape_aspect_relationship FOR
      relating_shape_aspect;

```



```
END_ENTITY; -- derived_shape_aspect
```

```
ENTITY derived_unit;
  elements : SET [1:?] OF derived_unit_element;
  WHERE
    WR1: ((SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (
      elements[1].exponent <> 1)));
END_ENTITY; -- derived_unit
```

```
ENTITY derived_unit_element;
  unit      : named_unit;
  exponent  : REAL;
END_ENTITY; -- derived_unit_element
```

```
ENTITY descriptive_representation_item
  SUBTYPE OF (representation_item);
  description : text;
END_ENTITY; -- descriptive_representation_item
```

```
ENTITY design_make_from_relationship
  SUBTYPE OF (product_definition_relationship);
  WHERE
    WR1: (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT'] * TYPEOF(SELF.relying_product_definition))
      = 1);
    WR2: (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT'] * TYPEOF(SELF.related_product_definition))
      = 1);
    WR3: (SIZEOF(QUERY ( pds <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_SHAPE.DEFINITION') IN TYPEOF(pds)) )) =
      0);
END_ENTITY; -- design_make_from_relationship
```

```
ENTITY design_object
  SUBTYPE OF (characterized_object);
  WHERE
```

```

        WR1: (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
END_ENTITY; -- design_object

```

```

ENTITY device_terminal_map
    SUBTYPE OF (shape_aspect, shape_aspect_relationship);
    UNIQUE
        UR1 : SELF\shape_aspect_relationship.related_shape_aspect,
            SELF\shape_aspect_relationship.relate_shape_aspect;
    WHERE
        WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART_TERMINAL') IN TYPEOF(SELF.
            relate_shape_aspect));
END_ENTITY; -- device_terminal_map

```

```

ENTITY dimension_related_tolerance_zone_element;
    related_dimension : dimensional_location;
    related_element   : tolerance_zone_definition;
END_ENTITY; -- dimension_related_tolerance_zone_element

```

```

ENTITY dimensional_characteristic_representation;
    dimension      : dimensional_characteristic;
    representation : shape_dimension_representation;
END_ENTITY; -- dimensional_characteristic_representation

```

```

ENTITY dimensional_exponents;
    length_exponent      : REAL;
    mass_exponent        : REAL;
    time_exponent        : REAL;
    electric_current_exponent : REAL;
    thermodynamic_temperature_exponent : REAL;
    amount_of_substance_exponent : REAL;
    luminous_intensity_exponent : REAL;
END_ENTITY; -- dimensional_exponents

```

```

ENTITY dimensional_location
    SUPERTYPE OF (ONEOF (angular_location, dimensional_location_with_path))
    SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- dimensional_location

```

```

ENTITY dimensional_location_with_direction
    SUBTYPE OF (dimensional_location);
    WHERE

```

```

        WR1: (SELF\shape_aspect_relationship.description = 'linear');
END_ENTITY; -- dimensional_location_with_direction

```

```

ENTITY dimensional_location_with_path
    SUBTYPE OF (dimensional_location);
    path : shape_aspect;
END_ENTITY; -- dimensional_location_with_path

```

```

ENTITY dimensional_size
    SUPERTYPE OF (angular_size);
    applies_to : shape_aspect;
    name       : label;
    WHERE
        WR1: (applies_to.product_definitional = TRUE);
END_ENTITY; -- dimensional_size

```

```

ENTITY dimensional_size_property
    SUBTYPE OF (dimensional_size, property_definition);
END_ENTITY; -- dimensional_size_property

```

```

ENTITY directed_action
    SUBTYPE OF (executed_action);
    directive : action_directive;
END_ENTITY; -- directed_action

```

```

ENTITY directed_dimensional_location
    SUBTYPE OF (dimensional_location);
END_ENTITY; -- directed_dimensional_location

```

```

ENTITY direction
    SUBTYPE OF (geometric_representation_item);
    direction_ratios : LIST [2:3] OF REAL;
    WHERE
        WR1: (SIZEOF(QUERY ( tmp <* direction_ratios | (tmp <> 0) )) > 0);
END_ENTITY; -- direction

```

```

ENTITY discrete_shield
    SUBTYPE OF (component_definition);
    WHERE
        WR1: (SELF.frame_of_reference.name = 'physical occurrence');
        WR2: (SIZEOF(QUERY ( si <* QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'shielded item') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(si.
        relating_product_definition)) )) >= 1);
END_ENTITY; -- discrete_shield

```

```

ENTITY document;
    id          : identifier;
    name        : label;
    description : text;
    kind        : document_type;
    UNIQUE
    UR1 : id;
END_ENTITY; -- document

```

```

ENTITY document_reference
    ABSTRACT SUPERTYPE;
    assigned_document : document;
    source            : label;
END_ENTITY; -- document_reference

```

```

ENTITY document_relationship;
    name          : label;
    description    : text;
    relating_document : document;
    related_document : document;
END_ENTITY; -- document_relationship

```

```

ENTITY document_type;
    product_data_type : label;
END_ENTITY; -- document_type

```

```

ENTITY document_usage_constraint;
    source          : document;
    subject_element : label;
    subject_element_value : text;
END_ENTITY; -- document_usage_constraint

```

```

ENTITY document_with_class
    SUBTYPE OF (document);
    class : identifier;
END_ENTITY; -- document_with_class

```

```

ENTITY edge
  SUPERTYPE OF (ONEOF (edge_curve,oriented_edge))
  SUBTYPE OF (topological_representation_item);
  edge_start : vertex;
  edge_end   : vertex;
END_ENTITY; -- edge

ENTITY edge_based_2d_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1 : (SIZEOF(QUERY ( it <* SELF\representation.items | (NOT (
      SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D'] * TYPEOF(it)) = 1)) )) = 0);
    WR2 : (SIZEOF(QUERY ( it <* SELF\representation.items | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM'] * TYPEOF(it)) = 1) )) >= 1);
    WR3 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
      items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( edges <* eb.ces_edges
        | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_CURVE') IN TYPEOF(edges))) )) = 0)) )) = 0)) )) = 0);
    WR4 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
      items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( pline_edges <*
        QUERY ( edges <* eb.ces_edges | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POLYLINE') IN TYPEOF(edges\edge_curve.edge_geometry)) ) |
        (NOT (SIZEOF(pline_edges\edge_curve.edge_geometry\polyline.
        points) > 2)) )) = 0)) )) = 0)) )) = 0);
    WR5 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
      items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( edges <* eb.ces_edges
        | (NOT ((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VERTEX_POINT') IN TYPEOF(edges.edge_start)) AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VERTEX_POINT') IN TYPEOF(edges.edge_end)))) )) = 0)) )) =
        0)) )) = 0);
WR6 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
        items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( edges <* eb.ces_edges
        | (NOT valid_2d_wireframe_edge_curve(edges\edge_curve.
        edge_geometry,

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
    ) ))
        = 0)) )) = 0)) )) = 0);
WR7 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
        items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( edges <* eb.ces_edges
        | (NOT (valid_wireframe_vertex_point(edges.edge_start\
        vertex_point.vertex_geometry) AND
        valid_wireframe_vertex_point(edges.edge_end\vertex_point.
        vertex_geometry)))) )) = 0)) )) = 0)) )) = 0);
WR8 : (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF\representation.
        items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( eb <* ebwm\edge_based_wireframe_model.
        ebwm_boundary | (NOT (SIZEOF(QUERY ( con_edges <*
        QUERY ( edges <* eb.ces_edges | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CONIC') IN TYPEOF(edges\edge_curve.edge_geometry)) ) | (
        NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D') IN TYPEOF(con_edges\edge_curve.
        edge_geometry\conic.position)))) )) = 0)) )) = 0)) )) = 0);
WR9 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF\representation.items
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM') IN TYPEOF(it)) ) | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(
        mi\mapped_item.mapping_source.mapped_representation)))) )) =
        0);
WR10: (SELF\representation.context_of_items\
        geometric_representation_context.coordinate_space_dimension
        = 2);
END_ENTITY; -- edge_based_2d_wireframe_shape_representation

```

```

ENTITY edge_based_wireframe_model
  SUBTYPE OF (geometric_representation_item);
  ebwm_boundary : SET [1:?] OF connected_edge_set;
END_ENTITY; -- edge_based_wireframe_model

```

```

ENTITY edge_based_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_3D']
      * TYPEOF(it)) = 1)) )) = 0);
    WR2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM']
      * TYPEOF(it)) = 1) )) >= 1);
    WR3: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL'
      IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
      edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
      QUERY ( edges <* eb.ces_edges | (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_CURVE'
      IN TYPEOF(edges))) )) = 0)) )) = 0)) )) = 0);
    WR4: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL'
      IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
      edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
      QUERY ( pline_edges <* QUERY ( edges <* eb.ces_edges | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'POLYLINE'
      IN TYPEOF(edges\edge_curve.edge_geometry))) )) | (NOT (SIZEOF(
      pline_edges\edge_curve.edge_geometry\polyline.points) > 2)) ))
      = 0)) )) = 0)) )) = 0);
    WR5: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_BASED_WIREFRAME_MODEL'
      IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
      edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
      QUERY ( edges <* eb.ces_edges | (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'VERTEX_POINT'

```

```

        IN TYPEOF(edges.edge_start)) AND (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VERTEX_POINT'
        IN TYPEOF(edges.edge_end)))))) = 0)) = 0)) = 0)) = 0);
WR6: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
        edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
        QUERY ( edges <* eb.ces_edges | (NOT
        valid_wireframe_edge_curve(edges\edge_curve.edge_geometry)) ))
        = 0)) = 0)) = 0)) = 0);
WR7: (SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
        edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
        QUERY ( edges <* eb.ces_edges | (NOT (
        valid_wireframe_vertex_point(edges.edge_start\vertex_point.
        vertex_geometry) AND valid_wireframe_vertex_point(edges.
        edge_end\vertex_point.vertex_geometry))) )) = 0)) = 0)) = 0)) = 0);
WR8: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM'
        IN TYPEOF(it)) ) | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation))) )) = 0);
WR9: (SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3);
END_ENTITY; -- edge_based_wireframe_shape_representation

```

```

ENTITY edge_curve
  SUBTYPE OF (edge, geometric_representation_item);
  edge_geometry : curve;
  same_sense : BOOLEAN;
END_ENTITY; -- edge_curve

```

```

ENTITY edge_loop
  SUBTYPE OF (loop, path);
  DERIVE
    ne : INTEGER := SIZEOF(SELF\path.edge_list);
  WHERE
    WR1: (SELF\path.edge_list[1].edge_start :=: SELF\path.edge_list[ne].
        edge_end);
END_ENTITY; -- edge_loop

```



```

ENTITY edge_segment_vertex
  SUBTYPE OF (physical_unit_datum);
  WHERE
    WR1: (SIZEOF(TYPEOF(SELF)) = 3);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
      'shape element characterization') )) = 1);
    WR3: (SELF\shape_aspect.of_shape\property_definition.definition.
      frame_of_reference\application_context_element.name IN [
      'physical design occurrence', 'physical design usage']);
    WR4: (SELF\shape_aspect.name = 'single datum');
END_ENTITY; -- edge_segment_vertex

ENTITY ee_specification
  SUBTYPE OF (document);
  WHERE
    WR1: ((SIZEOF(QUERY ( apoa <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa.
      role.name = 'document source') )) + SIZEOF(QUERY ( apoa <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa.role.name =
      'document source') ))) >= 1);
    WR2: (SELF\document.kind.product_data_type IN [
      'assembly technology specification', 'design specification',
      'fabrication technology specification',
      'interface specification', 'language reference manual',
      'lead form specification', 'material specification',
      'process specification', 'surface finish specification',
      'test specification']);
END_ENTITY; -- ee_specification

ENTITY effectivity
  SUPERTYPE OF (ONEOF (serial_numbered_effectivity, dated_effectivity,
    lot_effectivity));
  id : identifier;
END_ENTITY; -- effectivity

ENTITY electric_current_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ELECTRIC_CURRENT_UNIT'
      IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- electric_current_measure_with_unit

```

```

ENTITY electric_current_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 1) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- electric_current_unit

```

```

ENTITY electrical_network
  SUBTYPE OF (functional_unit);
END_ENTITY; -- electrical_network

```

```

ENTITY elementary_surface
  SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
    spherical_surface,toroidal_surface))
  SUBTYPE OF (surface);
  position : axis2_placement_3d;
END_ENTITY; -- elementary_surface

```

```

ENTITY ellipse
  SUBTYPE OF (conic);
  semi_axis_1 : positive_length_measure;
  semi_axis_2 : positive_length_measure;
END_ENTITY; -- ellipse

```

```

ENTITY evaluated_degenerate_pcurve
  SUBTYPE OF (degenerate_pcurve);
  equivalent_point : cartesian_point;
END_ENTITY; -- evaluated_degenerate_pcurve

```

```

ENTITY executed_action
  SUBTYPE OF (action);
END_ENTITY; -- executed_action

```

```

ENTITY expanded_uncertainty
  SUBTYPE OF (standard_uncertainty);
  coverage_factor : REAL;
END_ENTITY; -- expanded_uncertainty

```

```

ENTITY extension
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    WR1: (SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1);
END_ENTITY; -- extension

```

```

ENTITY external_definition
  SUBTYPE OF (characterized_object, externally_defined_item);
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
END_ENTITY; -- external_definition

```

```

ENTITY external_source;
  source_id : source_item;
END_ENTITY; -- external_source

```

```

ENTITY external_source_relationship;
  name          : label;
  description    : text;
  relating_source : external_source;
  related_source  : external_source;
END_ENTITY; -- external_source_relationship

```

```

ENTITY externally_defined_assembly_definition
  SUBTYPE OF (externally_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'assembly module') )) >= 1);
    WR2: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sa <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(sa)) AND (sa.
      description = 'pca terminal')) )) >= 2)) )) = 0)));
END_ENTITY; -- externally_defined_assembly_definition

```

```

ENTITY externally_defined_bare_die
  SUBTYPE OF (externally_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( ifdu <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATED_PRODUCT_DEFINITION') | (pdr.name =
      'implemented function') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT') IN TYPEOF(ifdu.
      relating_product_definition)) AND (ifdu.
      relating_product_definition.frame_of_reference.name =
      'functional design usage')) ) = 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( dut <*
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | (pdr.name = 'device unit technology') ) | (dut.
      relating_property_definition.name = 'unit technology') )) =
      1) )) = 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sa <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0);
  END_ENTITY; -- externally_defined_bare_die

```

```

ENTITY externally_defined_curve_font
  SUBTYPE OF (externally_defined_item);
  END_ENTITY; -- externally_defined_curve_font

```

```

ENTITY externally_defined_functional_unit
  SUBTYPE OF (functional_unit, externally_defined_item);
  END_ENTITY; -- externally_defined_functional_unit

```

```

ENTITY externally_defined_hatch_style
  SUBTYPE OF (externally_defined_item, geometric_representation_item);
  END_ENTITY; -- externally_defined_hatch_style

```

```

ENTITY externally_defined_interconnect_definition
  SUBTYPE OF (externally_defined_physical_unit);
  WHERE

```

```

WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
    product_category.name = 'interconnect module') )) >= 1);
END_ENTITY; -- externally_defined_interconnect_definition

```

```

ENTITY externally_defined_item;
    item_id : source_item;
    source : external_source;
END_ENTITY; -- externally_defined_item

```

```

ENTITY externally_defined_package
    SUBTYPE OF (externally_defined_physical_unit);
    WHERE
        WR1: (SELF.frame_of_reference.name = 'physical design usage');
        WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( pdr <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'registered case style') )) >= 1)) ))
            = 0);
        WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_REPRESENTATION') IN TYPEOF(pdr)) ) | (sr_pdr.
            used_representation.name = 'seating plane') )) = 1)) )) = 0);
        WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( sa <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_BODY') IN TYPEOF(sa)) )) <= 1)) )) = 0);
        WR5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( sa <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_TERMINAL') IN TYPEOF(sa)) )) >= 1)) )) = 0);
        WR6: ((NOT (SELF\product_definition.description = 'altered package'))
            OR (SIZEOF(QUERY ( pdr <* USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((pdr.name = 'package preparation') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE'] * TYPEOF(pdr.
        relating_product_definition)) = 1)) )) = 1));
END_ENTITY; -- externally_defined_package

ENTITY externally_defined_packaged_connector
    SUBTYPE OF (externally_defined_packaged_part);
END_ENTITY; -- externally_defined_packaged_connector

ENTITY externally_defined_packaged_part
    SUPERTYPE OF (externally_defined_packaged_connector)
    SUBTYPE OF (externally_defined_physical_unit);
    WHERE
        WR1: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( sa <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0));
        WR2: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MATERIAL_DESIGNATION.DEFINITIONS')) = 1));
        WR3: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(QUERY ( ifu <* QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | (pdr.name = 'implemented function') ) | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FUNCTIONAL_UNIT') IN TYPEOF(ifu.relatng_product_definition))
            AND (ifu.relatng_product_definition.frame_of_reference.name
            = 'functional design usage')) )) = 1));
        WR4: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(QUERY ( upkg <* QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
            | (pdr.name = 'used package') ) | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE') IN TYPEOF(upkg.related_product_definition)) )) =
            1));

```

```

WR5: ((NOT (SELF\product_definition.description =
    'altered packaged part')) OR (SIZEOF(QUERY ( bpp <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | (pdr.name = 'base packaged part') ) | ((SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(bpp.
    relating_product_definition)) = 1) AND (bpp.
    relating_product_definition.frame_of_reference.name =
    'physical design usage')) )) >= 1));
WR6: ((NOT (SELF\product_definition.description =
    'altered packaged part')) OR (SIZEOF(QUERY ( upkg <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | (pdr.name = 'used package') ) | (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGED_PART',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(upkg.
    relating_product_definition)) = 1) )) >= 1));
END_ENTITY; -- externally_defined_packaged_part

ENTITY externally_defined_physical_unit
    SUPERTYPE OF (library_defined_physical_unit ANDOR ONEOF (
        externally_defined_package,externally_defined_packaged_part,
        externally_defined_bare_die,externally_defined_assembly_definition,
        externally_defined_interconnect_definition))
    SUBTYPE OF (physical_unit, externally_defined_item);
END_ENTITY; -- externally_defined_physical_unit

ENTITY externally_defined_representation_item
    SUBTYPE OF (externally_defined_item, representation_item);
    WHERE
        WR1: (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
END_ENTITY; -- externally_defined_representation_item

ENTITY externally_defined_symbol
    SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_symbol

```

```

ENTITY externally_defined_text_font
  SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_text_font

```

```

ENTITY externally_defined_tile
  SUBTYPE OF (externally_defined_item);
END_ENTITY; -- externally_defined_tile

```

```

ENTITY externally_defined_tile_style
  SUBTYPE OF (externally_defined_item, geometric_representation_item);
END_ENTITY; -- externally_defined_tile_style

```

```

ENTITY extruded_area_solid
  SUBTYPE OF (swept_area_solid);
  extruded_direction : direction;
  depth              : positive_length_measure;
  WHERE
    WR1: (dot_product(SELF\swept_area_solid.swept_area.basis_surface\
      elementary_surface.position.p[3],extruded_direction) <> 0);
END_ENTITY; -- extruded_area_solid

```

```

ENTITY extruded_face_solid
  SUBTYPE OF (swept_face_solid);
  extruded_direction : direction;
  depth              : positive_length_measure;
  WHERE
    WR1: (dot_product(SELF\swept_face_solid.swept_face.face_geometry\
      elementary_surface.position.p[3],extruded_direction) <> 0);
END_ENTITY; -- extruded_face_solid

```

```

ENTITY fabrication_joint
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( ff <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'fabrication features') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL') IN TYPEOF(ff.related_shape_aspect))
      AND (ff.related_shape_aspect.description IN ['via terminal',
      'printed component join terminal',
      'non functional land join terminal','land join terminal',
      'conductive interconnect element terminal',
      'component termination passage join terminal']))) )) = 2);

```



```

WR2: (SIZEOF(QUERY ( ajm <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'auxiliary joint material') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND') IN TYPEOF(ajm.related_shape_aspect)) OR (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT') IN TYPEOF(ajm.related_shape_aspect))
    AND (ajm.related_shape_aspect.description =
    'stratum feature template component')))) )) <= 1);
END_ENTITY; -- fabrication_joint

```

```

ENTITY face
  SUPERTYPE OF (ONEOF (face_surface,oriented_face))
  SUBTYPE OF (topological_representation_item);
  bounds : SET [1:?] OF face_bound;
  WHERE
    WR1: (NOT mixed_loop_type_set(list_to_set(list_face_loops(SELF))));
    WR2: (SIZEOF(QUERY ( temp <* bounds | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FACE_OUTER_BOUND'
      IN TYPEOF(temp)) )) <= 1);
END_ENTITY; -- face

```

```

ENTITY face_bound
  SUBTYPE OF (topological_representation_item);
  bound      : loop;
  orientation : BOOLEAN;
END_ENTITY; -- face_bound

```

```

ENTITY face_outer_bound
  SUBTYPE OF (face_bound);
END_ENTITY; -- face_outer_bound

```

```

ENTITY face_surface
  SUBTYPE OF (face, geometric_representation_item);
  face_geometry : surface;
  same_sense    : BOOLEAN;
END_ENTITY; -- face_surface

```

```

ENTITY faceted_brep
  SUBTYPE OF (manifold_solid_brep);
END_ENTITY; -- faceted_brep

```

```

ENTITY faceted_brep_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FACETED_BREP',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT_3D']
        * TYPEOF(it)) = 1)) )) = 0);
    WR2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FACETED_BREP',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MAPPED_ITEM']
        * TYPEOF(it)) = 1) )) > 0);
    WR3: (SIZEOF(QUERY ( fbrep <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FACETED_BREP'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb_shells(
          fbrep,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
        | (NOT (SIZEOF(QUERY ( fcs <* csh.
          cfs_faces | (NOT ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FACE_SURFACE'
              IN TYPEOF(fcs)) AND (
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'PLANE'
                  IN TYPEOF(fcs\face_surface.face_geometry)) AND (
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'CARTESIAN_POINT'
                      IN TYPEOF(fcs\face_surface.face_geometry\elementary_surface.
                        position.location)))))) )) = 0)) )) = 0)) )) = 0);
    WR4: (SIZEOF(QUERY ( fbrep <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FACETED_BREP'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb_shells(
          fbrep,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
        | (NOT (SIZEOF(QUERY ( fcs <* csh.
          cfs_faces | (NOT (SIZEOF(QUERY ( bnds <* fcs.bounds | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FACE_OUTER_BOUND'
              IN TYPEOF(bnds)) )) = 1)) )) = 0)) )) = 0)) )) = 0);
    WR5: (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MANIFOLD_SOLID_BREP'
        IN TYPEOF(it)) ) | (
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'ORIENTED_CLOSED_SHELL'

```

```

        IN TYPEOF(msb\manifold_solid_brep.outer)) )) = 0);
WR6: (SIZEOF(QUERY ( brv <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BREP_WITH_VOIDS'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* brv\
        brep_with_voids.voids | csh\oriented_closed_shell.
        orientation )) = 0)) )) = 0);
WR7: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'
        IN TYPEOF(it)) ) | (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACETED_BREP_SHAPE_REPRESENTATION'
        IN TYPEOF(mi\mapped_item.mapping_source.
        mapped_representation)))) )) = 0);
END_ENTITY; -- faceted_brep_shape_representation

ENTITY fiducial
    SUBTYPE OF (component_shape_aspect);
    WHERE
        WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
        AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition.
        formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
        product_category.name = 'interconnect module') )) >= 1));
END_ENTITY; -- fiducial

ENTITY fiducial_part_feature
    SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- fiducial_part_feature

ENTITY fiducial_stratum_feature
    SUBTYPE OF (stratum_feature);
END_ENTITY; -- fiducial_stratum_feature

ENTITY fill_area_style;
    name : label;
    fill_styles : SET [1:?] OF fill_style_select;
    WHERE
        WR1: (SIZEOF(QUERY ( fill_style <* SELF.fill_styles | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FILL_AREA_STYLE_COLOUR') IN TYPEOF(fill_style)) )) <= 1);
END_ENTITY; -- fill_area_style

```

```

ENTITY fill_area_style_colour;
    name          : label;
    fill_colour   : colour;
END_ENTITY; -- fill_area_style_colour

```

```

ENTITY fill_area_style_hatching
    SUBTYPE OF (geometric_representation_item);
    hatch_line_appearance      : curve_style;
    start_of_next_hatch_line   : one_direction_repeat_factor;
    point_of_reference_hatch_line : cartesian_point;
    pattern_start              : cartesian_point;
    hatch_line_angle           : plane_angle_measure;
END_ENTITY; -- fill_area_style_hatching

```

```

ENTITY fill_area_style_tile_coloured_region
    SUBTYPE OF (geometric_representation_item);
    closed_curve : curve_or_annotation_curve_occurrence;
    region_colour : colour;
END_ENTITY; -- fill_area_style_tile_coloured_region

```

```

ENTITY fill_area_style_tile_curve_with_style
    SUBTYPE OF (geometric_representation_item);
    styled_curve : annotation_curve_occurrence;
END_ENTITY; -- fill_area_style_tile_curve_with_style

```

```

ENTITY fill_area_style_tile_symbol_with_style
    SUBTYPE OF (geometric_representation_item);
    symbol : annotation_symbol_occurrence;
END_ENTITY; -- fill_area_style_tile_symbol_with_style

```

```

ENTITY fill_area_style_tiles
    SUBTYPE OF (geometric_representation_item);
    tiling_pattern : two_direction_repeat_factor;
    tiles          : SET [1:?] OF fill_area_style_tile_shape_select;
    tiling_scale   : positive_ratio_measure;
END_ENTITY; -- fill_area_style_tiles

```

```

ENTITY founded_item;
END_ENTITY; -- founded_item

```

```

ENTITY functional_terminal_group
    SUBTYPE OF (group);

```

```

UNIQUE
  UR1 : SELF\group.name;
WHERE
  WR1: (SIZEOF(QUERY ( aga <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(aga)) AND (SIZEOF(
    QUERY ( dri <* aga.items | (dri.name =
    'functional group type') )) = 1)) )) = 1);
END_ENTITY; -- functional_terminal_group

ENTITY functional_unit
  SUPERTYPE OF (externally_defined_functional_unit ANDOR
    electrical_network ANDOR thermal_network)
  SUBTYPE OF (product_definition);
WHERE
  WR1: (SELF.frame_of_reference.name IN ['functional design usage',
    'functional network design']);
  WR2: ((NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS') IN TYPEOF(
    SELF))) OR (SIZEOF(QUERY ( docs <* SELF\
    product_definition_with_associated_documents.
    documentation_ids | (docs.kind.product_data_type =
    'CAD filename') )) <= 1));
  WR3: ((SIZEOF(QUERY ( adta <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (adta.role.name
    = 'creation date') )) + SIZEOF(QUERY ( ada <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_ASSIGNMENT.ITEMS') | (ada.role.name =
    'creation date') ))) = 1);
  WR4: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1);
  WR5: ((SIZEOF(QUERY ( apoa <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa.
    role.name = 'creator') )) + SIZEOF(QUERY ( apoa <* USEDIN(
    SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa.role.name =
    'creator') ))) >= 1);
  WR6: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1);
  WR7: ((NOT (SELF.frame_of_reference.name =
    'functional network design')) OR (SIZEOF(QUERY ( du <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
        | (du.relatng_product_definition.frame_of_reference.name =
        'functional design usage') )) = 1));
WR8: ((NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'THERMAL_NETWORK',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ELECTRICAL_NETWORK'] * TYPEOF(SELF)) = 1)) OR (SELF.
        frame_of_reference.name = 'functional network design'));
END_ENTITY; -- functional_unit

```

ENTITY functional_unit_terminal_definition

SUBTYPE OF (shape_aspect);

WHERE

```

WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(SELF.of_shape.definition)) AND
        (SELF.of_shape.definition\product_definition.
        frame_of_reference.name = 'functional design usage')));
WR2: (SIZEOF(QUERY ( pd2 <* QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION') IN TYPEOF(pd)) ) | (SIZEOF(
        QUERY ( funtdba <* QUERY ( pdr <* USEDIN(pd2,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name =
        'functional unit network terminal definition bus assignment') )
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(funtdba.
        relating_property_definition.definition)) )) <= 1) )) <= 1);
WR3: (SIZEOF(QUERY ( pd2 <* QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION') IN TYPEOF(pd)) ) | (SIZEOF(
        QUERY ( funtdna <* QUERY ( pdr <* USEDIN(pd2,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name =
        'functional unit network terminal definition node assignment') )
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'NETWORK_NODE_DEFINITION') IN TYPEOF(funtdna.
        relating_property_definition.definition)) )) <= 1) )) <= 1);
END_ENTITY; -- functional_unit_terminal_definition

```

ENTITY functionally_defined_transformation;

```

        name          : label;
        description    : text;
END_ENTITY; -- functionally_defined_transformation

```

```

ENTITY geometric_alignment
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    WR1: (SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1);
END_ENTITY; -- geometric_alignment

```

```

ENTITY geometric_curve_set
  SUBTYPE OF (geometric_set);
  WHERE
    WR1: (SIZEOF(QUERY ( temp <* SELF\geometric_set.elements | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SURFACE'
      IN TYPEOF(temp)) )) = 0);
END_ENTITY; -- geometric_curve_set

```

```

ENTITY geometric_intersection
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    WR1: (SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1);
END_ENTITY; -- geometric_intersection

```

```

ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
  coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

```

```

ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (point,direction,vector,placement,
    cartesian_transformation_operator,curve,surface,edge_curve,
    face_surface,poly_loop,vertex_point,solid_model,boolean_result,
    sphere,right_circular_cone,right_circular_cylinder,torus,block,
    right_angular_wedge,half_space_solid,shell_based_surface_model,
    shell_based_wireframe_model,edge_based_wireframe_model,
    geometric_set))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    WR1: (SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (
      NOT (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'GEOMETRIC_REPRESENTATION_CONTEXT'
        IN TYPEOF(using_rep.context_of_items))) )) = 0);
END_ENTITY; -- geometric_representation_item

```

```

ENTITY geometric_set
  SUPERTYPE OF (geometric_curve_set)
  SUBTYPE OF (geometric_representation_item);
  elements : SET [1:?] OF geometric_set_select;
END_ENTITY; -- geometric_set

```

```

ENTITY geometric_tolerance;
  name : label;
  description : text;
  magnitude : measure_with_unit;
  toleranced_shape_aspect : shape_aspect;
  WHERE
    WR1: (magnitude.value_component >= 0);
END_ENTITY; -- geometric_tolerance

```

```

ENTITY geometric_tolerance_group
  SUBTYPE OF (property_definition);
  WHERE
    WR1: (SELF\property_definition.description IN [
      'separate requirement', 'simultaneous requirement']);
    WR2: ((NOT (SELF\property_definition.description =
      'separate requirement')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
      SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM') IN
      TYPEOF(pdr.related_property_definition)) AND (pdr.name =
      'group geometric tolerance')) )) >= 1));
    WR3: ((NOT (SELF\property_definition.description =
      'simultaneous requirement')) OR (SIZEOF(QUERY ( pdr <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE') IN TYPEOF(pdr.
      related_property_definition)) AND (pdr.name =
      'group geometric tolerance')) )) >= 2));
    WR4: ((NOT (SELF\property_definition.description =
      'separate requirement')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
      SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
  | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINEAR_PROFILE_TOLERANCE') IN TYPEOF(pdr.
    related_property_definition)) AND (pdr.
    related_property_definition.name =
    'linear profile refinement') AND (pdr.name =
    'group geometric tolerance')) )) = 0));
WR5: ((NOT (SELF\property_definition.description =
'separate requirement')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
  | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(pdr.
    related_property_definition)) AND (pdr.
    related_property_definition.name =
    'surface profile refinement') AND (pdr.name =
    'group geometric tolerance')) )) = 0));
WR6: ((NOT (SELF\property_definition.description =
'separate requirement')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
  | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POSITION_TOLERANCE') IN TYPEOF(pdr.
    related_property_definition)) AND (pdr.
    related_property_definition.name =
    'feature relating position') AND (pdr.name =
    'group geometric tolerance')) )) = 0));
WR7: ((NOT (SELF\property_definition.description =
'separate requirement')) OR (SIZEOF(QUERY ( pdr <* USEDIN(
SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
  | ((pdr.name = 'group geometric tolerance') AND (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINEAR_PROFILE_TOLERANCE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_PROFILE_TOLERANCE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITION_TOLERANCE'] * TYPEOF(pdr.
    related_property_definition)) > 0)) )) = 0));
END_ENTITY; -- geometric_tolerance_group

```

```

ENTITY geometric_tolerance_relationship;
  name                : label;
  description         : text;
  relating_geometric_tolerance : geometric_tolerance;

```

```

        related_geometric_tolerance : geometric_tolerance;
END_ENTITY; -- geometric_tolerance_relationship

```

```

ENTITY geometric_tolerance_with_specified_datum_system
  SUBTYPE OF (physical_unit_geometric_tolerance);
  WHERE
    WR1: (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANGULARITY_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'POSITION_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CIRCULAR_RUNOUT_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LINEAR_PROFILE_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SURFACE_PROFILE_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONCENTRICITY_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PARALLELISM_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PERPENDICULARITY_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SYMMETRY_TOLERANCE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TOTAL_RUNOUT_TOLERANCE'] * TYPEOF(SELF)) = 1);
    WR2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((pdr.related_property_definition.description =
      'datum system property with material conditions') OR (pdr.
      related_property_definition.description =
      'datum system property without material conditions')) AND (
      pdr.name = 'referenced datum system definition')) ) = 1);
    WR3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_GROUP') IN TYPEOF(pdr.
      related_property_definition)) AND (pdr.
      related_property_definition.description =
      'separate requirement') AND (pdr.name =
      'group geometric tolerance')) ) <= 1);
END_ENTITY; -- geometric_tolerance_with_specified_datum_system

```

```

ENTITY geometrically_bounded_2d_wireframe_representation
  SUBTYPE OF (shape_representation);
  WHERE

```

```

WR1: (SELF.context_of_items\geometric_representation_context.
      coordinate_space_dimension = 2);
WR2: (SIZEOF(QUERY ( item <* SELF.items | (NOT (SIZEOF(TYPEOF(item)
      * [
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_CURVE_SET',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM'])
      = 1)) )) = 0);
WR3: (SIZEOF(QUERY ( item <* SELF.items | (SIZEOF(TYPEOF(item) * [
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_CURVE_SET',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM'])
      = 1)) )) >= 1);
WR4: (SIZEOF(QUERY ( mi <* QUERY ( item <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM'
        IN TYPEOF(item)) ) | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION') IN
        TYPEOF(mi\mapped_item.mapping_source.mapped_representation)))) ))
      = 0);
WR5: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_CURVE_SET'
        IN TYPEOF(item)) ) | (NOT (SIZEOF(QUERY ( elem <* gcs\
        geometric_set.elements | (NOT (SIZEOF(TYPEOF(elem) * [
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'B_SPLINE_CURVE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CIRCLE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_CURVE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ELLIPSE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'OFFSET_CURVE_2D',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POINT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POLYLINE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TRIMMED_CURVE'])
      = 1)) )) = 0)) )) = 0);
WR6: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_CURVE_SET'
        IN TYPEOF(item)) ) | (NOT (SIZEOF(QUERY ( crv <*
        QUERY ( elem <* gcs\geometric_set.elements | (

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CURVE'
    IN TYPEOF(elem)) ) | (NOT valid_basis_curve_in_2d_wireframe(
        crv)) )) = 0)) )) = 0);
WR7: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_CURVE_SET'
        IN TYPEOF(item)) ) | (NOT (SIZEOF(QUERY ( pnt <*
            QUERY ( elem <* gcs\geometric_set.elements | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POINT'
        IN TYPEOF(elem)) ) | (NOT (SIZEOF(TYPEOF(pnt)) * [
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_POINT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POINT_ON_CURVE'])
        = 1)) )) = 0)) )) = 0);
WR8: (SIZEOF(QUERY ( gcs <* QUERY ( item <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_CURVE_SET'
        IN TYPEOF(item)) ) | (NOT (SIZEOF(QUERY ( pl <*
            QUERY ( elem <* gcs\geometric_set.elements | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POLYLINE'
        IN TYPEOF(elem)) ) | (NOT (SIZEOF(pl\polyline.points) > 2)) ))
        = 0)) )) = 0);
END_ENTITY; -- geometrically_bounded_2d_wireframe_representation

```

```

ENTITY geometrically_bounded_surface_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
        WR1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_SET',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'AXIS2_PLACEMENT_3D'])
            * TYPEOF(it)) = 1)) )) = 0);
        WR2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_SET',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM'])
            * TYPEOF(it)) = 1)) > 0);
        WR3: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM'
                IN TYPEOF(it)) ) | (NOT ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION') IN

```

```

        TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
        = 0);
WR4: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( pnt <*
        QUERY ( gsel <* gs\geometric_set.elements | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POINT'
        IN TYPEOF(gsel)) ) | (NOT gbsf_check_point(pnt)) )) = 0)) ))
        = 0);
WR5: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cv <*
        QUERY ( gsel <* gs\geometric_set.elements | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE'
        IN TYPEOF(gsel)) ) | (NOT gbsf_check_curve(cv)) )) = 0)) ))
        = 0);
WR6: (SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( sf <*
        QUERY ( gsel <* gs\geometric_set.elements | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE'
        IN TYPEOF(gsel)) ) | (NOT gbsf_check_surface(sf)) )) = 0)) ))
        = 0);
END_ENTITY; -- geometrically_bounded_surface_shape_representation

```

```

ENTITY geometrically_bounded_wireframe_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
        WR1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF(TYPEOF(it) * [
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_CURVE_SET',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'AXIS2_PLACEMENT_3D',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM'])
            = 1)) )) = 0);
        WR2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF(TYPEOF(it) * [
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_CURVE_SET',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MAPPED_ITEM'])
            = 1)) )) >= 1);
        WR3: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_CURVE_SET'

```

```

        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( crv <*
        QUERY ( elem <* gcs\geometric_set.elements | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CURVE'
        IN TYPEOF(elem)) ) | (NOT
        valid_geometrically_bounded_wf_curve(crv,

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
) )) = 0)) )) = 0);
WR4: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_CURVE_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( pnts <*
        QUERY ( elem <* gcs\geometric_set.elements | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT'
        IN TYPEOF(elem)) ) | (NOT
        valid_geometrically_bounded_wf_point(pnts,

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
) )) = 0)) )) = 0);
WR5: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_CURVE_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cnc <*
        QUERY ( elem <* gcs\geometric_set.elements | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC'
        IN TYPEOF(elem)) ) | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_3D'
        IN TYPEOF(cnc\conic.position))) )) = 0)) )) = 0);
WR6: (SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRIC_CURVE_SET'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( pline <*
        QUERY ( elem <* gcs\geometric_set.elements | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE'
        IN TYPEOF(elem)) ) | (NOT (SIZEOF(pline\polyline.points) > 2)) ))
        = 0)) )) = 0);
WR7: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MAPPED_ITEM'
        IN TYPEOF(it)) ) | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION') IN
        TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
        = 0);
WR8: (SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3);
END_ENTITY; -- geometrically_bounded_wireframe_shape_representation

```

```

ENTITY global_uncertainty_assigned_context
  SUBTYPE OF (representation_context);
    uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY; -- global_uncertainty_assigned_context

```

```

ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
    units : SET [1:?] OF unit;
END_ENTITY; -- global_unit_assigned_context

```

```

ENTITY group;
  name      : label;
  description : text;
END_ENTITY; -- group

```

```

ENTITY group_assignment
  ABSTRACT SUPERTYPE;
  assigned_group : group;
END_ENTITY; -- group_assignment

```

```

ENTITY group_product_definition
  SUBTYPE OF (component_definition);
  WHERE
    WR1: (SELF.frame_of_reference.name = 'design requirement');
    WR2: ((NOT (SELF.description = 'placement group')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) )) >= 1)));
END_ENTITY; -- group_product_definition

```

```

ENTITY group_relationship;
  name      : label;
  description : text;
  relating_group : group;
  related_group : group;
END_ENTITY; -- group_relationship

```

```

ENTITY group_shape_aspect
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'GROUP_PRODUCT_DEFINITION') IN TYPEOF(SELF\shape_aspect.
        of_shape.definition));
WR2: ((NOT (SELF\shape_aspect.description =
        'interconnect module constraint region')) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUP_PRODUCT_DEFINITION') IN TYPEOF(SELF\shape_aspect.
        of_shape.definition)) AND (SIZEOF(QUERY ( gm <*
        QUERY ( pdr <* USEDIN(SELF\shape_aspect.of_shape.definition,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
        | (pdr.name = 'group member') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT') IN TYPEOF(gm.related_product_definition))
        AND (gm.related_product_definition.frame_of_reference.name =
        'physical design') AND (SIZEOF(QUERY ( prpc <* USEDIN(gm.
        related_product_definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
        product_category.name = 'interconnect module') )) >= 1)) ))
        = 1))) );
WR3: ((NOT (SELF\shape_aspect.description =
        'interconnect module constraint region')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.name = 'constrained object') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY') IN TYPEOF(pdr.
        related_property_definition.definition))) )) = 1) )) = 1));
WR4: ((NOT (SELF\shape_aspect.description =
        'interconnect module constraint region')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
        QUERY ( it <* pdr.used_representation.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'design specific purpose')) )) = 1) )) = 1)) )) = 0));
WR5: ((NOT (SELF\shape_aspect.description =
        'interconnect module constraint region')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(

```



```

        QUERY ( it <* pdr.used_representation.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'keepout') AND (it.description IN ['true','false'])) ))
        = 1) )) = 1)) )) = 0));
WR6: ((NOT (SELF\shape_aspect.description =
        'interconnect module constraint region')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
        = 1) )) = 1));
WR7: ((NOT (SELF\shape_aspect.description IN [
        'interconnect module constraint region',
        'termination constraint','placement group'])) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) )) = 1));
WR8: ((NOT (SELF\shape_aspect.description = 'termination constraint'))
        OR (SIZEOF(QUERY ( ctm <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'constrained termination member') ) | (ctm.
        related_shape_aspect.description =
        'mating connector termination') )) >= 2));
END_ENTITY; -- group_shape_aspect

```

```

ENTITY grouped_requirements_property
  SUBTYPE OF (group, requirements_property);
  WHERE
    WR1: ((NOT (SELF\group.name =
        'item restricted requirements property')) OR (SIZEOF(
        QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | (SIZEOF(aga.
        items) = 1) )) = 1));
    WR2: ((NOT (SELF\group.name = 'layout spacing requirements property'))
        OR (SIZEOF(QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | ((SIZEOF(aga.
        items) = 2) AND (SIZEOF(QUERY ( rp <* QUERY ( it <* aga.

```

```

items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY') IN TYPEOF(it)) ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESIGN_OBJECT') IN TYPEOF(rp.definition)) AND (rp.
definition.name IN ['component feature category',
'assembly component category', 'cutout category',
'fill area category', 'inter stratum feature category',
'stratum feature category', 'via category',
'altered package category']))) )) = 1)) )) = 1));
WR3: ((NOT (SELF\group.name = 'layout spacing requirements property'))
OR (SIZEOF(QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | ((SIZEOF(aga.
items) = 2) AND (SIZEOF(QUERY ( rp <* QUERY ( it <* aga.
items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY') IN TYPEOF(it)) ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESIGN_OBJECT') IN TYPEOF(rp.definition)) AND (rp.
definition.description = 'dependent design object category')) ))
= 1)) )) = 1));
WR4: ((NOT (SELF\group.name = 'layout spacing requirements property'))
OR (SIZEOF(QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | ((SIZEOF(aga.
items) = 2) AND (SIZEOF(QUERY ( rp <* QUERY ( it <* aga.
items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY') IN TYPEOF(it)) ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESIGN_OBJECT') IN TYPEOF(rp.definition)) AND (rp.
definition.description = 'reference design object category')) ))
= 1)) )) = 1));
END_ENTITY; -- grouped_requirements_property

```

```

ENTITY half_space_solid
  SUBTYPE OF (geometric_representation_item);
    base_surface : surface;
    agreement_flag : BOOLEAN;
END_ENTITY; -- half_space_solid

```

```

ENTITY hyperbola
  SUBTYPE OF (conic);
    semi_axis : positive_length_measure;
    semi_imag_axis : positive_length_measure;

```

```
END_ENTITY; -- hyperbola
```

```
ENTITY integral_shield
  SUBTYPE OF (component_shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(pdr.
        relating_property_definition)) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(pdr.
        relating_property_definition))) AND (pdr.name =
        'shielded item')) )) >= 1);
END_ENTITY; -- integral_shield
```

```
ENTITY inter_stratum_feature
  SUBTYPE OF (component_shape_aspect);
  WHERE
    WR1 : (SELF.description IN ['bonded conductive base blind via',
      'buried via', 'component termination passage',
      'interfacial connection', 'non conductive base blind via',
      'plated conductive base blind via', 'plated cutout',
      'partially plated cutout', 'plated cutout edge segment',
      'partially plated interconnect module edge',
      'plated interconnect module edge segment',
      'plated interconnect module edge', 'unsupported passage',
      'cutout', 'physical connectivity interrupting cutout',
      'dielectric material passage', 'cutout edge segment',
      'interconnect module edge segment',
      'interconnect module edge']);
    WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
        QUERY ( it <* pdr.used_representation.items | (((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
          name = 'feature of size') AND (it.description IN ['true',
          'false']))) )) = 1) )) = 1) )) = 1);
    WR3 : (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'instantiated template') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
```

```

relating_shape_aspect)) AND (it.relatiing_shape_aspect.
description IN ['inter stratum feature template',
'via template','component termination passage template',
'unsupported passage template'])) )) = 1);
WR4 : ((NOT (SELF.description = 'cutout edge segment')) OR (SIZEOF(
QUERY ( cc <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'composed cutout') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE') IN TYPEOF(cc.relatiing_shape_aspect))
AND ((cc.relatiing_shape_aspect.description = 'cutout') OR (
cc.relatiing_shape_aspect.description =
'physical connectivity interrupting cutout') OR (cc.
relatiing_shape_aspect.description =
'partially plated cutout')))) )) = 1));
WR5 : ((NOT (SELF.description = 'interconnect module edge segment'))
OR (SIZEOF(QUERY ( ce <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'composed edge') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE') IN TYPEOF(ce.relatiing_shape_aspect))
AND ((ce.relatiing_shape_aspect.description =
'interconnect module edge') OR (ce.relatiing_shape_aspect.
description = 'interconnect module edge')))) )) = 1));
WR6 : ((NOT (SELF.description = 'dielectric material passage')) OR (
SIZEOF(QUERY ( pp <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'precedent passage') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE') IN TYPEOF(pp.relatiing_shape_aspect))
AND ((pp.relatiing_shape_aspect.description = 'cutout') OR (
pp.relatiing_shape_aspect.description =
'physical connectivity interrupting cutout') OR (pp.
relatiing_shape_aspect.description =
'partially plated cutout')))) )) = 1));
WR7 : ((NOT (SELF.description = 'dielectric material passage')) OR (
SIZEOF(QUERY ( rp <* QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) ) | (rp.name =
'feature material') )) <= 1) OR (SIZEOF(USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1));
WR8 : ((NOT (SELF.description =
'physical connectivity interrupting cutout')) OR (SIZEOF(
QUERY ( ice <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.

```

```

name = 'interrupted connectivity elements') ) | (ice.
relating_shape_aspect.name =
'conductive interconnect element') ) ) >= 1));
WR9 : ((NOT (SELF.description = 'unsupported passage')) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relying_shape_aspect.
description = 'unsupported passage template')) ) ) = 1));
WR10: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition
.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'interconnect module') ) ) >= 1));
WR11: ((NOT (SELF\shape_aspect.description = 'cutout edge segment'))
OR (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE') IN TYPEOF(it
.relying_shape_aspect)) ) ) = 1));
WR12: ((NOT (SELF\shape_aspect.description =
'plated cutout edge segment')) OR (SIZEOF(QUERY ( it <*
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE') IN TYPEOF(it
.relying_shape_aspect)) ) ) = 1));
WR13: ((NOT (SELF\shape_aspect.description =
'interconnect module edge segment')) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE') IN TYPEOF(it
.relying_shape_aspect)) ) ) = 1));
WR14: ((NOT (SELF\shape_aspect.description =
'plated interconnect module edge segment')) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'INTER_STRATUM_FEATURE_EDGE_SEGMENT_TEMPLATE') IN TYPEOF(it
        .relating_shape_aspect)) )) = 1));
END_ENTITY; -- inter_stratum_feature

```

```

ENTITY inter_stratum_feature_edge_segment_template
  SUBTYPE OF (part_template_definition, shape_aspect_relationship);
  WHERE
    WR1: (SIZEOF(QUERY ( ce <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar <>
      []) ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTER_STRATUM_FEATURE_EDGE_TEMPLATE') IN TYPEOF(ce.
      relating_shape_aspect)) )) = 1);
    WR2: ((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect));
    WR3: ((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
      shape_aspect_relationship.relying_shape_aspect));
    WR4: (SELF\shape_aspect_relationship.relying_shape_aspect :<>: SELF
      \shape_aspect_relationship.related_shape_aspect);
    WR5: (SIZEOF(TYPEOF(SELF)) = 4);
END_ENTITY; -- inter_stratum_feature_edge_segment_template

```

```

ENTITY inter_stratum_feature_edge_template
  SUBTYPE OF (part_template_definition);
  WHERE
    WR1: (SIZEOF(TYPEOF(SELF)) = 4);
    WR2: (SELF\shape_aspect.description =
      'inter stratum feature edge template');
END_ENTITY; -- inter_stratum_feature_edge_template

```

```

ENTITY interconnect_definition
  SUBTYPE OF (physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'interconnect module') )) >= 1);
    WR2: ((NOT (SELF.frame_of_reference.name = 'physical design')) OR (
      SIZEOF(QUERY ( du <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.' +
      'RELATING_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
      | ((SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTERCONNECT_DEFINITION',

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_INTERCONNECT_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_INTERCONNECT_DEFINITION'] * TYPEOF(du)) = 1)
AND (du.related_product_definition.frame_of_reference.name =
    'physical design usage') AND (SIZEOF(QUERY ( prpc <*
    USEDIN(du.related_product_definition.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
    product_category.name = 'interconnect module') )) >= 1)) ))
    = 1));
END_ENTITY; -- interconnect_definition

ENTITY interconnect_module_design_object_category
    SUBTYPE OF (characterized_object);
    WHERE
        WR1: (SELF\characterized_object.description IN ['cutout category',
            'fill area category','inter stratum feature category',
            'stratum feature category','via category']);
END_ENTITY; -- interconnect_module_design_object_category

ENTITY interconnect_module_edge_segment
    SUBTYPE OF (inter_stratum_feature, shape_aspect_relationship);
    WHERE
        WR1: (SELF.description IN ['plated interconnect module edge segment',
            'interconnect module edge segment']);
        WR2: (SIZEOF(QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'join implementation') ) | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relateing_shape_aspect))
            AND (ji.relateing_shape_aspect.name = 'inter stratum join')) ))
            <= 1);
        WR3: ((NOT (SELF.description =
            'plated interconnect module edge segment')) OR (SIZEOF(
            QUERY ( ce <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'composed edge') ) | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PLATED_INTER_STRATUM_FEATURE') IN TYPEOF(ce.
            relating_shape_aspect)) AND (ce.relateing_shape_aspect.
            description = 'plated interconnect module edge')) )) = 1));
        WR4: ((NOT (SELF.description = 'interconnect module edge segment'))
            OR (SIZEOF(QUERY ( ce <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'composed edge') ) | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'INTER_STRATUM_FEATURE') IN TYPEOF(ce.relying_shape_aspect))
    AND (ce.relying_shape_aspect.description =
        'interconnect module edge')) )) = 1));
WR5: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
    shape_aspect_relationship.related_shape_aspect));
WR6: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
    shape_aspect_relationship.relying_shape_aspect));
WR7: (SELF\shape_aspect_relationship.relying_shape_aspect :<=: SELF
    \shape_aspect_relationship.related_shape_aspect);
END_ENTITY; -- interconnect_module_edge_segment

```

```

ENTITY interconnect_module_edge_segment_surface
    SUBTYPE OF (shape_aspect, shape_aspect_relationship);
    WHERE
        WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
            shape_aspect_relationship.related_shape_aspect));
        WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EDGE_SEGMENT_VERTEX') IN TYPEOF(SELF\
            shape_aspect_relationship.relying_shape_aspect));
        WR3: (SELF\shape_aspect_relationship.relying_shape_aspect :<=: SELF
            \shape_aspect_relationship.related_shape_aspect);
        WR4: (SIZEOF(QUERY ( ce <= QUERY ( sar <= USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'composed surface') ) | (
            'interconnect module edge surface' = ce.
            relying_shape_aspect.description) )) = 1);
        WR5: (SIZEOF(TYPEOF(SELF)) = 3);
    END_ENTITY; -- interconnect_module_edge_segment_surface

```

```

ENTITY interconnect_module_stratum_based_terminal
    SUBTYPE OF (interconnect_module_terminal);
    END_ENTITY; -- interconnect_module_stratum_based_terminal

```

```

ENTITY interconnect_module_terminal
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT') IN TYPEOF(SELF.of_shape.definition)) AND (
            SELF.of_shape.definition\product_definition.
            frame_of_reference.name = 'physical design usage') AND (
            SIZEOF(QUERY ( prpc <= USEDIN(SELF.of_shape.definition.
            formation.of_product,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\

```



```

        product_category.name = 'interconnect module') )) >= 1));
WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
        >= 1) )) >= 1);
WR3: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'member connected terminal') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_CONNECTED_TERMINALS_DEFINITION') IN TYPEOF(mct.
        relating_shape_aspect)) )) <= 1);
WR4: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        related_shape_aspect.description = 'connection zone') )) = 1);
END_ENTITY; -- interconnect_module_terminal

ENTITY interface_component_definition
  SUBTYPE OF (component_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL') IN TYPEOF(sa)) AND (sa.
        description IN ['interconnect component interface terminal',
        'packaged connector component interface terminal']))) )) >= 1) ))
        >= 1);
END_ENTITY; -- interface_component_definition

ENTITY interface_mounted_join
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL') IN TYPEOF(SELF.
        related_shape_aspect));
    WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL') IN TYPEOF(SELF.relying_shape_aspect));
END_ENTITY; -- interface_mounted_join

```

```

ENTITY interfaced_group_component_definition
  SUBTYPE OF (assembly_group_component_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( gc <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
      | (pdr.name = 'group component') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTERFACE_COMPONENT_DEFINITION') IN TYPEOF(gc.
      related_product_definition)) )) >= 1);
END_ENTITY; -- interfaced_group_component_definition

ENTITY intersection_curve
  SUBTYPE OF (surface_curve);
  WHERE
    WR1: (SIZEOF(SELF\surface_curve.associated_geometry) = 2);
    WR2: (associated_surface(SELF\surface_curve.associated_geometry[1])
      <> associated_surface(SELF\surface_curve.associated_geometry
      [2]));
END_ENTITY; -- intersection_curve

ENTITY item_defined_transformation;
  name          : label;
  description    : text;
  transform_item_1 : representation_item;
  transform_item_2 : representation_item;
END_ENTITY; -- item_defined_transformation

ENTITY join_shape_aspect
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( cp <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'connected point') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LAYER_CONNECTION_POINT') IN TYPEOF(cp.related_shape_aspect)) ))
      >= 2);
    WR2: (SIZEOF(QUERY ( nt <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'network topology') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_NETWORK') IN TYPEOF(nt.relating_shape_aspect)) ))
      = 1);
    WR3: ((NOT (SELF.name = 'intra stratum join')) OR (SIZEOF(
      QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'join implementation') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE') IN TYPEOF(ji.related_shape_aspect)) AND (
ji.related_shape_aspect.description = 'conductor')) )) <= 1));
WR4: ((NOT (SELF.name = 'intra stratum join')) OR (SIZEOF(
QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'join implementation') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE') IN TYPEOF(ji.related_shape_aspect)) AND (
ji.related_shape_aspect.description =
'connected filled area')) )) <= 1));
WR5: ((NOT (SELF.name = 'inter stratum join')) OR (SIZEOF(
QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'join implementation') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_CONCEPT_RELATIONSHIP') IN TYPEOF(ji.
related_shape_aspect)) AND (ji.related_shape_aspect\
shape_aspect.description =
'physical network supporting stratum feature conductive join'))
OR ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_INTER_STRATUM_FEATURE') IN TYPEOF(ji.
related_shape_aspect)))) )) <= 1));
END_ENTITY; -- join_shape_aspect

```

ENTITY land

```

SUBTYPE OF (component_shape_aspect);
WHERE
  WR1 : (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition
.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1));
  WR2 : (SIZEOF(QUERY ( i_f <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(i_f.
relating_shape_aspect)) )) = 1);
  WR3 : (SIZEOF(QUERY ( aud <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'alternate usage definition') ) | ((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(aud.
        relating_shape_aspect)) )) <= 1);
WR4 : (SIZEOF(QUERY ( sfi <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'stratum feature implementation') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(sfi.relatating_shape_aspect)) ))
        = 1);
WR5 : ((NOT (SELF\shape_aspect.description IN ['functional land',
        'via dependent land','via and contact size dependent land',
        'component termination passage dependent land',
        'contact size dependent land',
        'component termination passage and contact size dependent land']))
        OR (SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(ac.related_shape_aspect))
        AND (ac.related_shape_aspect.description =
        'land interface terminal')) )) = 1));
WR6 : ((NOT (SELF\shape_aspect.description IN ['functional land',
        'via dependent land','via and contact size dependent land',
        'component termination passage dependent land',
        'contact size dependent land',
        'component termination passage and contact size dependent land']))
        OR (SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL') IN TYPEOF(ac.related_shape_aspect))
        AND (ac.related_shape_aspect.description =
        'land join terminal')) )) >= 1));
WR7 : ((NOT (SELF\shape_aspect.description IN ['non functional land',
        'via dependent non functional land',
        'via and contact size dependent non functional land',
        'component termination passage dependent non functional land',
        'contact size dependent non functional land',
        'component termination passage and contact size ' +
        'dependent non functional land',
        'unsupported passage dependent non functional land'])) OR (
        SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'associated component') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(ac.related_shape_aspect))
        AND (ac.related_shape_aspect.description =
        'non functional land interface terminal')) )) = 1));

```

```

WR8 : ((NOT (SELF\shape_aspect.description IN ['non functional land',
'via dependent non functional land',
'via and contact size dependent non functional land',
'component termination passage dependent non functional land',
'contact size dependent non functional land',
'component termination passage and contact size ' +
'dependent non functional land',
'unsupported passage dependent non functional land']))) OR (
SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL') IN TYPEOF(ac.related_shape_aspect)))
AND (ac.related_shape_aspect.description =
'non functional land join terminal')) ) >= 1));
WR9 : ((NOT (SELF\shape_aspect.description IN [
'component termination passage and contact size dependent land',
'component termination passage and contact size dependent
+ 'non functional land',
'component termination passage dependent land',
'component termination passage dependent non functional land'])))
OR (SIZEOF(QUERY ( rp <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'reference passage') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE') IN TYPEOF(rp.relater_shape_aspect))) AND
(rp.related_shape_aspect.description =
'component termination passage')) ) = 1));
WR10: ((NOT (SELF\shape_aspect.description IN [
'via and contact size dependent land',
'via and contact size dependent non functional land',
'via dependent land', 'via dependent non functional land'])))
OR (SIZEOF(QUERY ( rv <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'reference via') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PLATED_PASSAGE') IN TYPEOF(rv.relater_shape_aspect))) AND
(rv.related_shape_aspect.description IN ['buried via',
'interfacial connection', 'bonded conductive base blind via',
'non conductive base blind via',
'plated conductive base blind via']))) ) = 1));
WR11: ((NOT (SELF\shape_aspect.description =
'unsupported passage dependent non functional land')) OR (
SIZEOF(QUERY ( rp <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'reference passage') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTER_STRATUM_FEATURE') IN TYPEOF(rp.relater_shape_aspect)))

```

```

AND (rp.relying_shape_aspect.description =
'unsupported passage')) )) = 1));
WR12: ((NOT (SELF\shape_aspect.description IN [
'component termination passage dependent land',
'component termination passage dependent non functional land']))
OR (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
relating_shape_aspect)) AND (it.related_shape_aspect.
description =
'default component termination passage based')) ))
= 1));
WR13: ((NOT (SELF\shape_aspect.description IN [
'component termination passage and contact size dependent land',
'component termination passage and contact size dependent '
+ 'non functional land'])) OR (SIZEOF(QUERY ( it <*
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
relating_shape_aspect)) AND (it.related_shape_aspect.
description = (
'default attachment size and component termination ' +
'passage based')))) )) = 1));
WR14: ((NOT (SELF\shape_aspect.description IN [
'contact size dependent land',
'contact size dependent non functional land'])) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
relating_shape_aspect)) AND (it.related_shape_aspect.
description = 'default attachment size based')) )) = 1));
WR15: ((NOT (SELF\shape_aspect.description IN [
'via and contact size dependent land',
'via and contact size dependent non functional land'])) OR
(SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
relating_shape_aspect)) AND (it.related_shape_aspect.
description = 'default attachment size and via based')) ))
= 1));

```

```

WR16: ((NOT (SELF\shape_aspect.description IN ['via dependent land',
'via dependent non functional land'])) OR (SIZEOF(
  QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'instantiated template') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
    relating_shape_aspect)) AND (it.related_shape_aspect.
    description = 'default via based')) ) = 1));
WR17: ((NOT (SELF\shape_aspect.description = (
  'unsupported passage dependent ' + 'non functional land'))))
OR (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
  name = 'instantiated template') ) | (((
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(it.
  relating_shape_aspect)) AND (it.relatng_shape_aspect.
  description = 'default unsupported passage based')) ) = 1));
END_ENTITY; -- land

ENTITY land_physical_template
  SUBTYPE OF (part_template_definition);
  WHERE
    WR1 : (SIZEOF(QUERY ( ada <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (SIZEOF(
      QUERY ( duc <* USEDIN(ada.assigned_document,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc.subject_element
        = 'pre defined classification code') ) = 1) ) <= 1);
    WR2 : (SIZEOF(QUERY ( am <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'access mechanism') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LAND_TEMPLATE_TERMINAL') IN TYPEOF(am.related_shape_aspect)) )
      >= 2);
    WR3 : ((NOT (SELF.description IN ['default attachment size based',
      'default attachment size and component termination passage based',
      'default attachment size and via based'])) OR (SIZEOF(
      QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
        name = 'minimum attachment region size') AND (sar.
        related_shape_aspect.description = 'connection zone')) ) =
      1));
    WR4 : ((NOT (SELF.description IN ['default attachment size based',
      'default attachment size and component termination passage based',
      'default attachment size and via based'])) OR (SIZEOF(

```

```

QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'maximum attachment region size') AND (sar.
related_shape_aspect.description = 'connection zone')) )) =
1));
WR5 : ((NOT (SELF.description IN [
'default component termination passage based',
'default attachment size and component termination passage based']))
OR (SIZEOF(QUERY ( tu <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'technology usage') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY') IN TYPEOF(tu.related_shape_aspect))
AND (tu.related_shape_aspect.description =
'default component termination passage definition')) )) = 1));
WR6 : ((NOT (SELF.description IN [
'default attachment size and component termination passage based',
'default attachment size based',
'default attachment size and via based',
'default component termination passage based',
'default via based', 'default unsupported passage based']))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
| (pdr.name = 'technology usage') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_TECHNOLOGY') IN TYPEOF(tu.
relating_property_definition.definition)) )) = 1) )) = 1));
WR7 : ((NOT (SELF.description IN [
'default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based',
'default via based', 'default unsupported passage based']))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) )) = 2) )) =
1)) )) = 0));
WR8 : ((NOT (SELF.description IN [
'default attachment size and component termination passage based',

```



```

        'default attachment size and via based',
        'default component termination passage based',
        'default via based', 'default unsupported passage based']))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
=
'minimum standard fabrication allowance')) )) = 1) )) = 1)) ))
= 0));
WR9 : ((NOT (SELF.description IN [
'default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based',
'default via based', 'default unsupported passage based']))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum annular ring')) )) = 1) )) = 1)) )) = 0));
WR10: ((NOT (SELF.description IN [
'default attachment size and component termination passage based',
'default attachment size and via based',
'default component termination passage based',
'default via based', 'default unsupported passage based']))
OR (SIZEOF(QUERY ( tu <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'technology usage') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY') IN TYPEOF(tu.related_shape_aspect)) ))
= 1));
WR11: ((NOT (SELF.description IN ['default via based',
'default attachment size and via based'])) OR (SIZEOF(
QUERY ( tu <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'technology usage') ) | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PASSAGE_TECHNOLOGY') IN TYPEOF(tu.related_shape_aspect))
    AND (tu.related_shape_aspect.description =
        'default via definition')) )) = 1));
WR12: ((NOT (SELF.description = 'default unsupported passage')) OR (
    SIZEOF(QUERY ( tu <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'technology usage') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PASSAGE_TECHNOLOGY') IN TYPEOF(tu.related_shape_aspect))
        AND (tu.related_shape_aspect.description =
            'default unsupported passage definition')) )) = 1));
END_ENTITY; -- land_physical_template

```

```

ENTITY land_template_terminal
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (SELF.description IN ['interface terminal', 'join terminal']);
        WR2: (SIZEOF(QUERY ( tcz <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
            name = 'terminal connection zone') AND (sar.description IN [
                'edge curve', 'edge point', 'surface area', 'surface point'])) )
            | (tcz.related_shape_aspect.description = 'connection zone') ))
            >= 1);
        WR3: (SIZEOF(QUERY ( tcz <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
            name = 'terminal connection zone') ) | (NOT (SIZEOF(
            QUERY ( tcz_1 <* (QUERY ( sar <* USEDIN(SELF,
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
                name = 'terminal connection zone') ) - tcz) | (NOT (tcz.
                description = tcz_1.description)) )) = 0)) )) = 0);
END_ENTITY; -- land_template_terminal

```

```

ENTITY layer
    SUBTYPE OF (shape_aspect);
    UNIQUE
        UR1 : SELF\shape_aspect.name;
    WHERE
        WR1: (SIZEOF(QUERY ( ada <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (SIZEOF(
            QUERY ( duc <* USEDIN(ada.assigned_document,
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc.subject_element =
                'layer definition') )) = 1) )) <= 1);
        WR2: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(SELF.of_shape.
        definition)) AND (SELF.of_shape.definition.name =
        'inter stratum extent'));
END_ENTITY; -- layer

ENTITY layer_connection_point
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF\shape_aspect.of_shape.
        definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
        product_category.name = 'design layer') )) >= 1);
    WR2: (SIZEOF(QUERY ( cp <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'connected point') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'JOIN_SHAPE_ASPECT') IN TYPEOF(cp.relatng_shape_aspect)) ))
        >= 1);
    WR3: ((NOT (SELF\shape_aspect.description = 'dependently located'))
        OR (SIZEOF(QUERY ( ado <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated design object') ) | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTER_STRATUM_FEATURE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_TERMINAL'] * TYPEOF(ado.
        relating_shape_aspect)) = 1) )) = 1));
    WR4: ((NOT (SELF\shape_aspect.description = 'explicitly located'))
        OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation))
        AND (pdr.used_representation.name =
        'connection point location') AND (SIZEOF(QUERY ( it <* pdr.
        used_representation.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CARTESIAN_POINT') IN TYPEOF(it)) )) = 1)) )) = 1)) )) = 0));
END_ENTITY; -- layer_connection_point

```

```

ENTITY layer_connection_point_link
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);

```

```

WHERE
  WR1: (SELF\shape_aspect_relationship.related_shape_aspect :<>: SELF\
    shape_aspect_relationship.relate_shape_aspect);
  WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYER_CONNECTION_POINT') IN TYPEOF(SELF\
    shape_aspect_relationship.related_shape_aspect));
  WR3: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LAYER_CONNECTION_POINT') IN TYPEOF(SELF\
    shape_aspect_relationship.relate_shape_aspect));
  WR4: (SIZEOF(TYPEOF(SELF)) = 3);
  WR5: (SELF\shape_aspect.name = '');
  WR6: (SELF\shape_aspect_relationship.name = '');
  WR7: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')) = 1);
END_ENTITY; -- layer_connection_point_link

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LENGTH_UNIT'
      IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- length_measure_with_unit

ENTITY length_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 1) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- length_unit

ENTITY library_defined_assembly_definition
  SUBTYPE OF (library_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'assembly module') )) >= 1);
    WR2: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(sa)) AND (sa.
        description = 'pca terminal')) )) >= 2)) )) = 0));
END_ENTITY; -- library_defined_assembly_definition

```

```

ENTITY library_defined_bare_die
  SUBTYPE OF (library_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( ifdu <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') | (pdr.name =
        'implemented function') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(ifdu.
        relating_product_definition)) AND (ifdu.
        relating_product_definition.frame_of_reference.name =
        'functional design usage')) )) = 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( dut <*
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'device unit technology') ) | (dut.
        relating_property_definition.name = 'unit technology') )) =
        1) )) = 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0);
END_ENTITY; -- library_defined_bare_die

```

```

ENTITY library_defined_functional_unit
  SUBTYPE OF (externally_defined_functional_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( esr <* USEDIN(SELF.source,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE') | (esr.name
        = 'revision') )) = 0);
END_ENTITY; -- library_defined_functional_unit

```

```

ENTITY library_defined_interconnect_definition
  SUBTYPE OF (library_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'interconnect module') )) >= 1);
END_ENTITY; -- library_defined_interconnect_definition

ENTITY library_defined_model
  SUBTYPE OF (externally_defined_item, analytical_model,
    product_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( esr <* USEDIN(SELF.source,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE') | (esr.name
      = 'revision') )) = 1);
END_ENTITY; -- library_defined_model

ENTITY library_defined_package
  SUBTYPE OF (library_defined_physical_unit);
  WHERE
    WR1: (SELF.frame_of_reference.name = 'physical design usage');
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
      used_representation.name = 'registered case style') )) >= 1)) ))
      = 0);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION') IN TYPEOF(pdr)) ) | (sr_pdr.
      used_representation.name = 'seating plane') )) = 1)) )) = 0);
    WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sa <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE_BODY') IN TYPEOF(sa)) )) <= 1)) )) = 0);
    WR5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_TERMINAL') IN TYPEOF(sa)) )) >= 1)) )) = 0);
WR6: ((NOT (SELF.description = 'altered package')) OR (SIZEOF(
        QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((pdr.name = 'package preparation') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE'] * TYPEOF(pdr.
        relating_product_definition)) = 1)) )) = 1));
END_ENTITY; -- library_defined_package

ENTITY library_defined_packaged_connector
    SUBTYPE OF (library_defined_packaged_part);
END_ENTITY; -- library_defined_packaged_connector

ENTITY library_defined_packaged_part
    SUBTYPE OF (library_defined_physical_unit);
    WHERE
        WR1: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( sa <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0));
        WR2: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MATERIAL_DESIGNATION.DEFINITIONS')) = 1));
        WR3: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
            OR (SIZEOF(QUERY ( ifu <* QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | (pdr.name = 'implemented function') ) | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FUNCTIONAL_UNIT') IN TYPEOF(ifu.relatng_product_definition))
            AND (ifu.relatng_product_definition.frame_of_reference.name
            = 'functional design usage')) )) = 1));

```

```

WR4: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
      OR (SIZEOF(QUERY ( upkg <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
        | (pdr.name = 'used package') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE') IN TYPEOF(upkg.related_product_definition)) )) =
      1));
WR5: ((NOT (SELF.description = 'altered packaged part')) OR (SIZEOF(
      QUERY ( bpp <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'base packaged part') ) | ((SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(bpp.
        relating_product_definition)) = 1) AND (bpp.
        relating_product_definition.frame_of_reference.name =
        'physical design usage')) )) >= 1));
WR6: ((NOT (SELF.description = 'altered packaged part')) OR (SIZEOF(
      QUERY ( upkg <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'used package') ) | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(upkg.
        relating_product_definition)) = 1) )) >= 1));
END_ENTITY; -- library_defined_packaged_part

```

```

ENTITY library_defined_physical_unit
  SUPERTYPE OF (ONEOF (library_defined_package,
    library_defined_assembly_definition,
    library_defined_interconnect_definition,
    library_defined_packaged_part, library_defined_bare_die))
  SUBTYPE OF (externally_defined_physical_unit);
  WHERE
    WR1: (SIZEOF(QUERY ( esr <* USEDIN(SELF.source,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE') | (esr.name
      = 'revision') )) = 0);
END_ENTITY; -- library_defined_physical_unit

```

ENTITY line


```

SUBTYPE OF (curve);
  pnt : cartesian_point;
  dir : vector;
WHERE
  WR1: (dir.dim = pnt.dim);
END_ENTITY; -- line

```

```

ENTITY linear_profile_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
  WHERE
    WR1: ((NOT (SELF\geometric_tolerance.name =
      'linear profile refinement')) OR (SIZEOF(QUERY ( gtr < *
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
        | (((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'LINEAR_PROFILE_TOLERANCE') IN TYPEOF(gtr.
            relating_geometric_tolerance)) AND (gtr.name =
              'linear profile refining control') AND ((gtr.
                relating_geometric_tolerance.name =
                  'linear profile locating') OR (gtr.
                    relating_geometric_tolerance.name =
                      'linear profile refinement')))) = 1)));
    WR2: ((NOT (SELF\geometric_tolerance.name =
      'linear profile locating')) OR (SIZEOF(QUERY ( gtr < *
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
        | (((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'LINEAR_PROFILE_TOLERANCE') IN TYPEOF(gtr.
            relating_geometric_tolerance)) AND (gtr.
              relating_geometric_tolerance.name =
                'linear profile refinement') AND (gtr.name =
                  'linear profile refining control')) = 1)));
    WR3: ((NOT (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM') IN
      TYPEOF(SELF)) AND (SELF\geometric_tolerance.name =
        'linear profile refinement')) OR (SIZEOF(QUERY ( gtr < *
          USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
          | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LINEAR_PROFILE_TOLERANCE') IN TYPEOF(gtr.
              relating_geometric_tolerance)) AND (gtr.
                relating_geometric_tolerance.name =
                  'linear profile refinement') AND (gtr.name =
                    'linear profile refining control')) = 0)));

```

```

WR4: ((NOT (SELF\geometric_tolerance.name =
        'linear profile refinement')) OR (SIZEOF(QUERY ( gtr < *
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LINEAR_PROFILE_TOLERANCE') IN TYPEOF(gtr.
        relating_geometric_tolerance)) AND (gtr.
        related_geometric_tolerance.name =
        'linear profile refinement') AND (gtr.name =
        'linear profile refining control')) )) <= 1));
WR5: (NOT (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF)));
END_ENTITY; -- linear_profile_tolerance

```

```

ENTITY local_time;
    hour_component      : hour_in_day;
    minute_component    : OPTIONAL minute_in_hour;
    second_component    : OPTIONAL second_in_minute;
    zone                : coordinated_universal_time_offset;
WHERE
    WR1: valid_time(SELF);
END_ENTITY; -- local_time

```

```

ENTITY loop
    SUPERTYPE OF (ONEOF (vertex_loop, edge_loop, poly_loop))
    SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

```

```

ENTITY lot_effectivity
    SUBTYPE OF (effectivity);
    effectivity_lot_id   : identifier;
    effectivity_lot_size : measure_with_unit;
END_ENTITY; -- lot_effectivity

```

```

ENTITY luminous_intensity_measure_with_unit
    SUBTYPE OF (measure_with_unit);
WHERE
    WR1: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LUMINOUS_INTENSITY_UNIT'
        IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- luminous_intensity_measure_with_unit

```

```

ENTITY luminous_intensity_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 1));
END_ENTITY; -- luminous_intensity_unit

ENTITY make_from_connectivity_relationship
  SUBTYPE OF (shape_aspect_relationship);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(SELF.
      relating_shape_aspect));
    WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(SELF.
      related_shape_aspect));
END_ENTITY; -- make_from_connectivity_relationship

ENTITY make_from_feature_relationship
  SUBTYPE OF (shape_aspect_relationship);
  WHERE
    WR1: (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_INTERFACE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT'] * TYPEOF(SELF.
      relating_shape_aspect)) = 1);
    WR2: (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_INTERFACE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT'] * TYPEOF(SELF.related_shape_aspect))
      = 1);
    WR3: ((NOT ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT') IN TYPEOF(SELF.
      relating_shape_aspect))) OR (SELF.relatating_shape_aspect.
      description IN ['component feature',
      'component termination passage interface terminal',
      'land or non functional land interface terminal',

```

```

        'printed connector component interface terminal']));
WR4: ((NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_SHAPE_ASPECT') IN TYPEOF(SELF.
    related_shape_aspect))) OR (SELF.related_shape_aspect.
    description IN ['component feature',
    'component termination passage interface terminal',
    'land or non functional land interface terminal',
    'printed connector component interface terminal']));
END_ENTITY; -- make_from_feature_relationship

```

```

ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
  ranking          : INTEGER;
  ranking_rationale : text;
  quantity         : measure_with_unit;
  WHERE
    WR1: (ranking > 0);
END_ENTITY; -- make_from_usage_option

```

```

ENTITY manifold_solid_brep
  SUBTYPE OF (solid_model);
  outer : closed_shell;
END_ENTITY; -- manifold_solid_brep

```

```

ENTITY manifold_surface_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_SURFACE_MODEL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D']
    * TYPEOF(it)) = 1)) )) = 0);
    WR2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_SURFACE_MODEL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM']
    * TYPEOF(it)) = 1) )) > 0);
    WR3 : (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'
    IN TYPEOF(it)) ) | (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MANIFOLD_SURFACE_SHAPE_REPRESENTATION'

```

```

        IN TYPEOF(mi\mapped_item.mapping_source.
        mapped_representation))) ) = 0);
WR4 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_SURFACE_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( sh <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'OPEN_SHELL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ORIENTED_CLOSED_SHELL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CLOSED_SHELL']
        * TYPEOF(sh)) = 1)) )) = 0)) )) = 0);
WR5 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_SURFACE_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT (
        SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACE_SURFACE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ORIENTED_FACE']
        * TYPEOF(fa)) = 1)) )) = 0)) )) = 0)) )) = 0);
WR6 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_SURFACE_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( f_sf <* QUERY ( fa <* cfs\connected_face_set.
        cfs_faces | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACE_SURFACE'
        IN TYPEOF(fa)) ) | (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ADVANCED_FACE'
        IN TYPEOF(f_sf)) OR (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'B_SPLINE_SURFACE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELEMENTARY_SURFACE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'OFFSET_SURFACE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE_REPLICA',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SWEPT_SURFACE']
        * TYPEOF(f_sf\face_surface.face_geometry)) = 1))) )) = 0)) ))
        = 0)) )) = 0);
WR7 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR msf_surface_check(fa\face_surface.
    face_geometry))) )) = 0)) )) = 0)) )) = 0);
WR8 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( bnds <* fa.bounds | (NOT
    (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'])
    * TYPEOF(bnds.bound)) = 1)) )) = 0))) )) = 0)) )) = 0)) ))
    = 0);
WR9 : (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( elp_fbnds <*
    QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds\path.edge_list | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_CURVE'
    IN TYPEOF(oe.edge_element))) )) = 0)) )) = 0))) )) = 0)) ))
    = 0)) )) = 0);
WR10: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( elp_fbnds <*

```

```

        QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
        IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe_cv <*
        QUERY ( oe <* elp_fbnds\path.edge_list | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_CURVE'
        IN TYPEOF(oe.edge_element)) ) | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'B_SPLINE_CURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CURVE_REPLICA',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'OFFSET_CURVE_3D',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PCURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_CURVE']
        * TYPEOF(oe_cv.edge_element\edge_curve.edge_geometry)) = 1)) ))
        = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
WR11: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
        IN TYPEOF(fa)) OR (SIZEOF(QUERY ( elp_fbnds <*
        QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
        IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
        elp_fbnds\path.edge_list | (NOT msf_curve_check(oe.
        edge_element\edge_curve.edge_geometry)) )) = 0)) )) = 0))) ))
        = 0)) )) = 0)) )) = 0);
WR12: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
        IN TYPEOF(fa)) OR (SIZEOF(QUERY ( elp_fbnds <*
        QUERY ( bnds <* fa.bounds | (

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds\path.edge_list | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(oe.edge_element.edge_start)) AND (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(oe.edge_element.edge_end)))))) = 0))) = 0))) ))
    = 0))) = 0))) = 0));
WR13: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( elp_fbnds <*
    QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds\path.edge_list | (NOT ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DEGENERATE_PCURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_CURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_SURFACE']
    * TYPEOF(oe.edge_element.edge_start\vertex_point.
    vertex_geometry)) = 1) AND (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DEGENERATE_PCURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_CURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_SURFACE']
    * TYPEOF(oe.edge_element.edge_end\vertex_point.
    vertex_geometry)) = 1)))) = 0))) = 0))) = 0))) = 0))) = 0)))
    = 0));
WR14: (SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
    shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((

```



```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( vlp_fbnds <*
    QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
    IN TYPEOF(vlp_fbnds\vertex_loop.loop_vertex))) )) = 0))) ))
    = 0)) )) = 0)) )) = 0);
WR15: (SIZEOF(QUERY ( sbms <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbms\
    shell_based_surface_model.sbms_boundary | (NOT (SIZEOF(
    QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ADVANCED_FACE'
    IN TYPEOF(fa)) OR (SIZEOF(QUERY ( vlp_fbnds <*
    QUERY ( bnds <* fa.bounds | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'
    IN TYPEOF(bnds.bound)) ) | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DEGENERATE_PCURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_CURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POINT_ON_SURFACE']
    * TYPEOF(vlp_fbnds\vertex_loop.loop_vertex\vertex_point.
    vertex_geometry)) = 1)) )) = 0))) )) = 0)) )) = 0)) )) = 0);
END_ENTITY; -- manifold_surface_shape_representation

```

```

ENTITY mapped_item
  SUBTYPE OF (representation_item);
    mapping_source : representation_map;
    mapping_target : representation_item;
  WHERE
    WR1: acyclic_mapped_representation(using_representations(SELF),[SELF]);
END_ENTITY; -- mapped_item

```

```

ENTITY mass_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MASS_UNIT'

```

```

        IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- mass_measure_with_unit

```

```

ENTITY mass_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 1) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- mass_unit

```

```

ENTITY material_designation;
  name      : label;
  definitions : SET [1:?] OF characterized_definition;
END_ENTITY; -- material_designation

```

```

ENTITY material_designation_characterization;
  name      : label;
  description : text;
  designation : material_designation;
  property   : characterized_material_property;
END_ENTITY; -- material_designation_characterization

```

```

ENTITY material_electrical_conductivity_category
  SUBTYPE OF (group);
  WHERE
    WR1: (SELF.name IN ['conductive', 'non conductive', 'resistive',
      'semi conductive']);
END_ENTITY; -- material_electrical_conductivity_category

```

```

ENTITY material_property
  SUBTYPE OF (property_definition);
  UNIQUE
  UR1 : SELF\property_definition.name, SELF\property_definition.definition;
  WHERE
    WR1: ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CHARACTERIZED_OBJECT'
      IN TYPEOF(SELF\property_definition.definition)) OR (SIZEOF(
      bag_to_set(USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
QUERY ( temp <* bag_to_set(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_PROPERTY_REPRESENTATION') IN TYPEOF(temp)) )) = 0));
END_ENTITY; -- material_property

```

```

ENTITY material_property_representation
  SUBTYPE OF (property_definition_representation);
  dependent_environment : data_environment;
END_ENTITY; -- material_property_representation

```

```

ENTITY measure_qualification;
  name : label;
  description : text;
  qualified_measure : measure_with_unit;
  qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    WR1: (SIZEOF(QUERY ( temp <* qualifiers | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRECISION_QUALIFIER'
      IN TYPEOF(temp)) )) < 2);
END_ENTITY; -- measure_qualification

```

```

ENTITY measure_representation_item
  SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item

```

```

ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (length_measure_with_unit, mass_measure_with_unit,
    time_measure_with_unit, electric_current_measure_with_unit,
    thermodynamic_temperature_measure_with_unit,
    amount_of_substance_measure_with_unit,
    luminous_intensity_measure_with_unit, plane_angle_measure_with_unit,
    solid_angle_measure_with_unit, area_measure_with_unit,
    volume_measure_with_unit, ratio_measure_with_unit));
  value_component : measure_value;
  unit_component : unit;
  WHERE
    WR1: valid_units(SELF);
END_ENTITY; -- measure_with_unit

```

```

ENTITY minimally_defined_bare_die_terminal

```

```

SUPERTYPE OF (bare_die_terminal)
SUBTYPE OF (shape_aspect);
WHERE
  WR1: (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BARE_DIE' ) IN TYPEOF(SELF.of_shape.definition));
  WR2: (SIZEOF(QUERY ( eca <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT' ) | (sar.
    name = 'external connection area' ) ) | (eca.
    related_shape_aspect.description = 'connection zone' ) ) ) <=
    1);
  WR3: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_DESIGNATION.DEFINITIONS' ) ) <= 1);
  WR4: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT' ) | (sar.
    name = 'member connected terminal' ) ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION' ) IN TYPEOF(mct.
    relating_shape_aspect)) ) ) <= 1);
END_ENTITY; -- minimally_defined_bare_die_terminal

ENTITY model_parameter
  SUBTYPE OF (representation_item);
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS' ) ) = 1);
    WR2: (SIZEOF(QUERY ( cri <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION.ITEMS' ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COORDINATED_REPRESENTATION_ITEM' ) IN TYPEOF(cri)) AND (cri\
      representation.name =
      'model parameter with valid range value' ) ) ) <= 1);
    WR3: (SIZEOF(QUERY ( aga <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT.ITEMS' ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PARAMETER_TYPE' ) IN TYPEOF(aga.assigned_group)) AND (aga.
      assigned_group.name IN ['string property type',
      'logical property type', 'physical property type',
      'boolean property type' ] ) ) ) = 1);
  END_ENTITY; -- model_parameter

ENTITY modified_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
  modifier : limit_condition;
  END_ENTITY; -- modified_geometric_tolerance

```

```

ENTITY mounting_restriction_area
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
      AND (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT') IN TYPEOF(sar.related_shape_aspect))
      AND (sar.name = 'mounting surface') AND (sar.
      related_shape_aspect.of_shape.definition = sar.
      relating_shape_aspect.of_shape.definition)) )) = 1));
    WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(sar.
      relating_shape_aspect.of_shape.definition)) AND (sar.
      relating_shape_aspect.of_shape.definition.name =
      'inter stratum extent')) )) >= 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
      = 1)) )) = 0);
    WR4: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT') IN TYPEOF(sar.related_shape_aspect))
      AND (sar.name = 'mounting surface') AND (sar.
      related_shape_aspect.description =
      'interconnect module component surface feature')) )) = 1);
END_ENTITY; -- mounting_restriction_area

```

```

ENTITY named_unit
  SUPERTYPE OF (ONEOF (si_unit,conversion_based_unit,
    context_dependent_unit) ANDOR ONEOF (length_unit,mass_unit,
    time_unit,electric_current_unit,thermodynamic_temperature_unit,
    amount_of_substance_unit,luminous_intensity_unit,plane_angle_unit,
    solid_angle_unit,area_unit,volume_unit,ratio_unit));
  dimensions : dimensional_exponents;
END_ENTITY; -- named_unit

```

```

ENTITY network_node_definition
  SUBTYPE OF (product_definition);
  WHERE
    WR1: (SELF.frame_of_reference.name = 'functional network design');
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(
        QUERY ( funtdna <* QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
          | (pdr.name =
            'functional unit network terminal definition node assignment') )
          | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(funtdna.
            related_property_definition)) AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FUNCTIONAL_UNIT_TERMINAL_DEFINITION') IN TYPEOF(funtdna.
            related_property_definition.definition))) ) <= 1) ) <= 1);
    WR3: (SIZEOF(QUERY ( funn <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
      | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(funn.
        relating_product_definition)) AND (funn.
        relating_product_definition.frame_of_reference.name =
        'functional network design')) ) = 1);
END_ENTITY; -- network_node_definition

```

```

ENTITY next_assembly_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY; -- next_assembly_usage_occurrence

```

```

ENTITY offset_curve_2d
  SUBTYPE OF (curve);
  basis_curve      : curve;
  distance          : length_measure;
  self_intersect   : LOGICAL;
  WHERE
    WR1: (basis_curve.dim = 2);
END_ENTITY; -- offset_curve_2d

```

```

ENTITY offset_curve_3d
  SUBTYPE OF (curve);
  basis_curve      : curve;
  distance          : length_measure;
  self_intersect   : LOGICAL;

```

```

    ref_direction : direction;
WHERE
    WR1: ((basis_curve.dim = 3) AND (ref_direction.dim = 3));
END_ENTITY; -- offset_curve_3d

ENTITY offset_surface
    SUBTYPE OF (surface);
    basis_surface : surface;
    distance      : length_measure;
    self_intersect : LOGICAL;
END_ENTITY; -- offset_surface

ENTITY one_direction_repeat_factor
    SUBTYPE OF (geometric_representation_item);
    repeat_factor : vector;
END_ENTITY; -- one_direction_repeat_factor

ENTITY open_shell
    SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell

ENTITY opposing_boundary_dimensional_size
    SUBTYPE OF (dimensional_size);
WHERE
    WR1: (SELF\dimensional_size.name IN ['angular', 'linear']);
END_ENTITY; -- opposing_boundary_dimensional_size

ENTITY organization;
    id : OPTIONAL identifier;
    name : label;
    description : text;
END_ENTITY; -- organization

ENTITY organization_assignment
    ABSTRACT SUPERTYPE;
    assigned_organization : organization;
    role : organization_role;
END_ENTITY; -- organization_assignment

ENTITY organization_relationship;
    name : label;
    description : text;

```

```

        relating_organization : organization;
        related_organization  : organization;
    END_ENTITY; -- organization_relationship

```

```

ENTITY organization_role;
    name : label;
END_ENTITY; -- organization_role

```

```

ENTITY organizational_address
    SUBTYPE OF (address);
    organizations : SET [1:?] OF organization;
    description   : text;
END_ENTITY; -- organizational_address

```

```

ENTITY organizational_project;
    name                : label;
    description          : text;
    responsible_organizations : SET [1:?] OF organization;
END_ENTITY; -- organizational_project

```

```

ENTITY oriented_closed_shell
    SUBTYPE OF (closed_shell);
    closed_shell_element : closed_shell;
    orientation          : BOOLEAN;
    DERIVE
        SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                                    conditional_reverse(SELF.
                                                    orientation,SELF.
                                                    closed_shell_element.cfs_faces);
    WHERE
        WR1: (NOT (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ORIENTED_CLOSED_SHELL'
            IN TYPEOF(SELF.closed_shell_element)));
END_ENTITY; -- oriented_closed_shell

```

```

ENTITY oriented_edge
    SUBTYPE OF (edge);
    edge_element : edge;
    orientation  : BOOLEAN;
    DERIVE
        SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
                                                    SELF.edge_element.edge_start,SELF.
                                                    edge_element.edge_end);
        SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,

```



```

SELF.edge_element.edge_end, SELF.
edge_element.edge_start);

WHERE
  WR1: (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ORIENTED_EDGE'
    IN TYPEOF(SELF.edge_element)));
END_ENTITY; -- oriented_edge

ENTITY oriented_face
  SUBTYPE OF (face);
  face_element : face;
  orientation : BOOLEAN;
  DERIVE
    SELF\face.bounds : SET [1:?] OF face_bound := conditional_reverse(
      SELF.orientation, SELF.face_element.bounds);
  WHERE
    WR1: (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ORIENTED_FACE'
      IN TYPEOF(SELF.face_element)));
END_ENTITY; -- oriented_face

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);
  open_shell_element : open_shell;
  orientation : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
      conditional_reverse(SELF.
        orientation, SELF.
        open_shell_element.cfs_faces);
  WHERE
    WR1: (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ORIENTED_OPEN_SHELL'
      IN TYPEOF(SELF.open_shell_element)));
END_ENTITY; -- oriented_open_shell

ENTITY oriented_path
  SUBTYPE OF (path);
  path_element : path;
  orientation : BOOLEAN;
  DERIVE
    SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
      conditional_reverse(SELF.orientation, SELF.
        path_element.edge_list);
  WHERE

```

```

    WR1: (NOT (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ORIENTED_PATH'
        IN TYPEOF(SELF.path_element)));
END_ENTITY; -- oriented_path

ENTITY outer_boundary_curve
    SUBTYPE OF (boundary_curve);
END_ENTITY; -- outer_boundary_curve

ENTITY package
    SUBTYPE OF (physical_unit);
    WHERE
        WR1 : (SELF.frame_of_reference.name = 'physical design usage');
        WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'registered case style') )) >= 1) ))
            >= 1);
        WR3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_BODY') IN TYPEOF(sa)) )) = 1) )) <= 1);
        WR4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_TERMINAL') IN TYPEOF(sa)) )) >= 1) )) >= 1);
        WR5 : ((NOT (SELF.description = 'altered package')) OR (SIZEOF(
            QUERY ( pdr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | ((pdr.name = 'package alteration') AND (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EXTERNALLY_DEFINED_PACKAGE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LIBRARY_DEFINED_PACKAGE'] * TYPEOF(pdr.
            relating_product_definition)) = 1)) )) = 1));

```

```

WR6 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
    used_representation.name = 'package mounting data') )) = 1) ))
    = 1);
WR7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
    used_representation.name = 'package mounting data') AND (
    SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | (item.name =
    'maximum body height above seating plane') )) = 1)) )) = 1) ))
    = 1);
WR8 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
    used_representation.name = 'package mounting data') AND (
    SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | (item.name =
    'maximum body height below seating plane') )) = 1)) )) = 1) ))
    <= 1);
WR9 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
    used_representation.name = 'package mounting data') AND (
    SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | (item.name =
    'maximum body clearance above seating plane') )) = 1)) )) =
    1) )) <= 1);
WR10: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
    used_representation.name = 'package mounting data') AND (
    SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (item.name =
        'maximum body clearance below seating plane') )) = 1)) )) =
        1) )) <= 1);
WR11: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
        used_representation.name = 'package mounting data') AND (
        SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (item.name =
        'minimum body clearance above seating plane') )) = 1)) )) =
        1) )) <= 1);
WR12: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
        used_representation.name = 'package mounting data') AND (
        SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (item.name =
        'minimum body clearance below seating plane') )) = 1)) )) =
        1) )) <= 1);
WR13: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
        used_representation.name = 'package mounting data') AND (
        SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (item.name =
        'maximum lead length below seating plane') )) = 1)) )) = 1) ))
        <= 1);
WR14: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
        used_representation.name = 'package mounting data') AND (
        SIZEOF(QUERY ( item <* USEDIN(pdr.used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (item.name =
        'least lead length below seating plane') )) = 1)) )) = 1) ))
        <= 1);

```

```

WR15: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'package seating plane') )) = 1) )) = 1) )) = 1);
WR16: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SEATING_PLANE') IN TYPEOF(sa)) )) = 1) )) = 1);
END_ENTITY; -- package

```

```

ENTITY package_body
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(TYPEOF(SELF.of_shape.definition) * [
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_PACKAGE']) = 1);
    WR2: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_DESIGNATION.DEFINITIONS')) = 1);
END_ENTITY; -- package_body

```

```

ENTITY package_body_bottom_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_bottom_surface

```

```

ENTITY package_body_surface
  ABSTRACT SUPERTYPE OF (ONEOF (package_body_top_surface,
    package_body_bottom_surface))
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(TYPEOF(SELF.of_shape.definition) * [
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PACKAGE',

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE']) = 1);
WR2: SELF\shape_aspect.product_definitional;
WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (TYPEOF(
        sar\shape_aspect_relationship.relateing_shape_aspect) = [
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY']) )) = 1);
END_ENTITY; -- package_body_surface

ENTITY package_body_top_surface
  SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_top_surface

ENTITY package_terminal
  SUPERTYPE OF (primary_reference_terminal)
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1 : (SELF\shape_aspect.description IN ['altered package terminal',
        'guided wave terminal','wire terminal',
        'primary reference terminal','']);
    WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
        >= 1)) )) = 0);
    WR3 : (SIZEOF(TYPEOF(SELF.of_shape.definition) * [
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE']) = 1);
    WR4 : (SIZEOF(QUERY ( md <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS') | (SIZEOF(
        QUERY ( pd <* QUERY ( defs <* md.definitions | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION') IN TYPEOF(defs)) ) | (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((pdr.name = 'material assembly relationship') AND (pdr.
        description = 'terminal core material') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

      'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(pdr))) )) = 1) )) = 1) ))
    = 1);
WR5 : (SIZEOF(QUERY ( md <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_DESIGNATION.DEFINITIONS') | (SIZEOF(
    QUERY ( pd <* QUERY ( defs <* md.definitions | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION') IN TYPEOF(defs)) ) | (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | ((pdr.name = 'material assembly relationship') AND (pdr.
    description = 'terminal surface material') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(pdr))) )) = 1) )) = 1) ))
    = 1);
WR6 : (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'member connected terminal') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_CONNECTED_TERMINALS_DEFINITION') IN TYPEOF(mct.
    relating_shape_aspect))) <= 1);
WR7 : ((NOT (SELF\shape_aspect.description =
    'altered package terminal')) OR (SIZEOF(QUERY ( mct <*
    QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name IN ['length trimmed preparation',
    'shape formed preparation', 'surface preparation'])) ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_TERMINAL') IN TYPEOF(mct.relateing_shape_aspect))) ))
    = 1));
WR8 : ((NOT (SELF\shape_aspect.description =
    'altered package terminal')) OR (SELF.of_shape.definition\
    product_definition.description = 'altered package'));
WR9 : ((NOT (SELF\shape_aspect.description =
    'length trimmed terminal')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( it <* pdr.used_representation.items | (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) )) = 2) )) =
    1) )) = 0));
WR10: ((NOT (SELF\shape_aspect.description =
    'length trimmed terminal')) OR (SIZEOF(QUERY ( pd <*

```

```

USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum trimmed length')) )) = 1) )) = 1)) )) = 0));
WR11: ((NOT (SELF\shape_aspect.description =
'length trimmed terminal')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum trimmed length')) )) = 1) )) = 1)) )) = 0));
WR12: ((NOT (SELF\shape_aspect.description = 'shape formed terminal'))
OR (SIZEOF(QUERY ( ada <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EE_SPECIFICATION') IN TYPEOF(ada.assigned_document)) AND (
ada.assigned_document.kind.product_data_type =
'lead form specification') AND (SIZEOF(QUERY ( dr <*
USEDIN(ada.assigned_document,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DOCUMENT_RELATIONSHIP.RELATED_DOCUMENT') | (dr.
relating_document.kind.product_data_type =
'material specification') )) = 1)) )) = 1));
WR13: ((NOT (SELF\shape_aspect.description =
'surface prepped terminal')) OR (SIZEOF(QUERY ( ada <*
USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EE_SPECIFICATION') IN TYPEOF(ada.assigned_document)) AND (
ada.assigned_document.kind.product_data_type =
'surface finish specification')) )) = 1));
WR14: ((NOT (SELF\shape_aspect.description = 'wire terminal')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```



```

QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_UNIT') IN TYPEOF(it\measure_with_unit.
unit_component)))) )) = 2) )) = 1)) )) = 0));
WR15: ((NOT (SELF\shape_aspect.description = 'wire terminal')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name
= 'maximum wire terminal length') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_UNIT') IN TYPEOF(it\measure_with_unit.
unit_component)))) )) = 1) )) = 1)) )) = 0));
WR16: ((NOT (SELF\shape_aspect.description = 'wire terminal')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.name
= 'minimum wire terminal length') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_UNIT') IN TYPEOF(it\measure_with_unit.
unit_component)))) )) = 1) )) = 1)) )) = 0));
WR17: ((NOT (SELF\shape_aspect.description = 'wire terminal')) OR (
SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'internal connection zone') AND (sar.
related_shape_aspect.description = 'connection zone')) )) =
1));
WR18: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'external connection zone') AND (sar.
related_shape_aspect.description = 'connection zone')) )) =
1);
WR19: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( lmwu <* QUERY ( it <* pdr.used_representation.items
| (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) ) | (lmwu.
name = 'maximum terminal diametrical extent')) <= 1) ))
<= 1) )) <= 1);
WR20: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( lmwu <* QUERY ( it <* pdr.used_representation.items
| (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) ) | (lmwu.
name = 'minimum terminal diametrical extent')) <= 1) ))
<= 1) )) <= 1);
END_ENTITY; -- package_terminal

```

```

ENTITY packaged_component
  SUBTYPE OF (component_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( ip <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
      | (pdr.name = 'instantiated part')) | ((SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PACKAGED_PART',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(ip.
      relating_product_definition)) = 1) AND (ip.
      relating_product_definition.frame_of_reference.name =
      'physical design usage')))) = 1);
    WR2: (SIZEOF(QUERY ( pa <* QUERY ( pdr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
      | (pdr.name = 'package alternate')) | ((SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'EXTERNALLY_DEFINED_PACKAGE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGE'] * TYPEOF(pa.
        relating_product_definition)) = 1) AND (pa.
        relating_product_definition.description = 'altered package')) ))
        <= 1);
WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL') IN TYPEOF(sa)) AND (sa.description =
        'packaged component join terminal')) )) >= 1)) )) = 0);
WR4: ((NOT (SELF.description = 'packaged connector component')) OR (
        SIZEOF(QUERY ( ip <* QUERY ( pdr <* USEDIN(SELf,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'instantiated part') ) | ((SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(ip.
        relating_product_definition)) = 1) AND (ip.
        relating_product_definition.frame_of_reference.name =
        'packaged connector')) )) = 1));
WR5: ((NOT (SELF.description = 'packaged connector component')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL') IN TYPEOF(sa)) AND (sa.description =
        'packaged component join terminal')) )) >= 1)) )) = 0));
WR6: ((NOT (SELF.description = 'routed packaged component')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(QUERY ( pd <* USEDIN(sa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1) )) =
        1) )) = 1)) )) = 0));
END_ENTITY; -- packaged_component

```

```

ENTITY packaged_connector
  SUBTYPE OF (packaged_part);
  WHERE
    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP') IN TYPEOF(sa)) ))
      = 1) )) >= 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(QUERY ( sar <* USEDIN(sa,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'interface plane') )) = 1) )) = 1) )) <= 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SEATING_PLANE') IN TYPEOF(sa)) )) = 1) )) <= 1);
  END_ENTITY; -- packaged_connector

```

```

ENTITY packaged_connector_terminal_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  WHERE
    WR1: ((SIZEOF(TYPEOF(SELF.of_shape.definition) * [
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_PACKAGED_PART',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_PACKAGED_PART']) = 1) AND (SELF.of_shape.
      definition\product_definition.description =
      'packaged connector'));
    WR2: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART_TERMINAL') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect)) AND (SELF\
      shape_aspect_relationship.related_shape_aspect.description =
      'interface terminal'));
    WR3: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART_TERMINAL') IN TYPEOF(SELF\

```

```

        shape_aspect_relationship.related_shape_aspect)) AND (SELF\
        shape_aspect_relationship.related_shape_aspect.description =
        'join terminal'));
END_ENTITY; -- packaged_connector_terminal_relationship

```

```

ENTITY packaged_part

```

```

    SUPERTYPE OF (packaged_connector)

```

```

    SUBTYPE OF (physical_unit);

```

```

    WHERE

```

```

        WR1: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
        OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0)));
        WR2: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
        OR (SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1)));
        WR3: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
        OR (SIZEOF(QUERY ( ifu <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'implemented function') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(ifu.relying_product_definition))
        AND (ifu.relying_product_definition.frame_of_reference.name
        = 'functional design usage')) )) = 1)));
        WR4: ((NOT (SELF.frame_of_reference.name = 'physical design usage'))
        OR (SIZEOF(QUERY ( upkg <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'used package') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE') IN TYPEOF(upkg.relying_product_definition)) )) =
        1)));
        WR5: ((NOT (SELF.description = 'altered packaged part')) OR (SIZEOF(
        QUERY ( bpp <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'base packaged part') ) | ((SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(bpp.
        relating_product_definition)) = 1) AND (bpp.

```

```

        relating_product_definition.frame_of_reference.name =
        'physical design usage')) )) >= 1));
WR6: ((NOT (SELF.description = 'altered packaged part')) OR (SIZEOF(
    QUERY ( upkg <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | (pdr.name = 'used package') ) | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(upkg.
        relating_product_definition)) = 1) )) >= 1));
END_ENTITY; -- packaged_part

```

```

ENTITY packaged_part_terminal
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (SELF\shape_aspect.description IN ['interface terminal',
            'join terminal']);
        WR2: ((SIZEOF(TYPEOF(SELF.of_shape.definition) * [
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EXTERNALLY_DEFINED_PACKAGED_PART',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LIBRARY_DEFINED_PACKAGED_PART']) = 1) AND (SELF.of_shape.
            definition\product_definition.frame_of_reference.name =
            'physical design usage'));
        WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( pdr <* USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
            >= 1)) )) = 0);
        WR4: (SIZEOF(QUERY ( top <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
            name = 'terminal of package') ) | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_TERMINAL') IN TYPEOF(top.related_shape_aspect)) ))
            = 1);
        WR5: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'member connected terminal') ) | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PART_CONNECTED_TERMINALS_DEFINITION') IN TYPEOF(mct.
        relating_shape_aspect)) )) <= 1);
WR6: ((NOT (SELF\shape_aspect.description = 'interface terminal'))
OR (SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP') IN TYPEOF(sar))
AND
(sar.relatng_shape_aspect.description = 'join terminal')) ))
>= 1));
WR7: ((NOT (SELF\shape_aspect.description = 'join terminal')) OR (
SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP') IN TYPEOF(sar))
AND (sar.relatng_shape_aspect.description =
'interface terminal')) )) <= 1));
END_ENTITY; -- packaged_part_terminal

```

```

ENTITY parabola
  SUBTYPE OF (conic);
  focal_dist : length_measure;
  WHERE
    WR1: (focal_dist <> 0);
END_ENTITY; -- parabola

```

```

ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
  WHERE
    WR1: (SELF\geometric_tolerance.name = 'parallelism');
END_ENTITY; -- parallelism_tolerance

```

```

ENTITY parameter_assignment_representation
  SUBTYPE OF (representation);
  WHERE
    WR1: (SIZEOF(SELF.items) = 2);
    WR2: (SIZEOF(QUERY ( it <* SELF.items | ((it.name =
      'assigned parameter') AND (SIZEOF(TYPEOF(it) * [
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MODEL_PARAMETER',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM']) = 1)) )) = 1);
END_ENTITY; -- parameter_assignment_representation

```

```

ENTITY parameter_type
  SUBTYPE OF (group);
  WHERE
    WR1: (SELF\group.name IN ['string property type',
      'logical property type', 'physical property type',
      'boolean property type']);
END_ENTITY; -- parameter_type

```

```

ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

```

```

ENTITY part_connected_terminals_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'member connected terminal') ) | (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_MODULE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'INTERCONNECT_MODULE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGE_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PACKAGED_PART_TERMINAL'] * TYPEOF(mct.related_shape_aspect))
      = 1) )) >= 2);
END_ENTITY; -- part_connected_terminals_definition

```

```

ENTITY part_mounting_feature
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
      name = 'mounting area') AND (sar.related_shape_aspect.
      description = 'connection zone')) )) = 1);
END_ENTITY; -- part_mounting_feature

```

```

ENTITY part_template_definition
  SUBTYPE OF (shape_aspect, externally_defined_item);

```


WHERE

```

WR1 : (SELF\shape_aspect.of_shape\property_definition.definition.
      frame_of_reference\application_context_element.name =
      'template definition');
WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANALYTICAL_MODEL') IN TYPEOF(pdr.used_representation)) AND
      (pdr.used_representation.name =
      'part template analytical model')) )) <= 1)) )) = 0);
WR3 : ((NOT (SELF.description =
      'component termination passage template')) OR (SIZEOF(
      QUERY ( ctpt <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'component termination passage technology') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PASSAGE_TECHNOLOGY') IN TYPEOF(ctpt.relate_shape_aspect))
      AND (ctpt.relate_shape_aspect.description =
      'default component termination passage definition')) )) = 1));
WR4 : ((NOT (SELF.description =
      'component termination passage template')) OR (SIZEOF(
      QUERY ( am <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'access mechanism') ) | (am.relate_shape_aspect.
      description IN [
      'component termination passage template interface terminal',
      'component termination passage template join terminal']) ))
      >= 2));
WR5 : ((NOT (SELF.description = 'default trace template')) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | (pdr.name = 'technology usage') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM_TECHNOLOGY') IN TYPEOF(tu.
      relate_property_definition.definition)) )) = 1) )) = 1));
WR6 : ((NOT (SELF.description = 'inter stratum feature template'))
      OR (SIZEOF(QUERY ( isfpt <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'inter stratum feature passage technology') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PASSAGE_TECHNOLOGY') IN TYPEOF(

```

```

        isfpt.relatering_shape_aspect))) ))
    = 1));
WR7 : ((NOT (SELF.description = 'printed connector template')) OR (
    SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name IN ['connector interface terminal',
        'connector join terminal']) )) >= 1));
WR8 : ((NOT (SELF.description =
    'printed part cross section template')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL') IN TYPEOF(pdr.used_representation)) AND
        (pdr.used_representation.name = 'transmission line model')) ))
    = 1)) )) = 0));
WR9 : ((NOT (SELF.description =
    'printed part cross section template')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( it <* pdr.used_representation.items | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) )) = 2) )) =
    1)) )) = 0));
WR10: ((NOT (SELF.description =
    'printed part cross section template')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
        = 'maximum width')) )) = 1) )) = 1)) )) = 0));
WR11: ((NOT (SELF.description =
    'printed part cross section template')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum width')) )) = 1) )) = 1)) )) = 0));
WR12: ((NOT (SELF.description =
'printed part cross section template')) OR (SIZEOF(
QUERY ( am <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'access mechanism') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_CROSS_SECTION_TEMPLATE_TERMINAL') IN TYPEOF(
am.related_shape_aspect))) >= 2));
WR13: ((NOT (SELF.description = 'printed part template')) OR (
SIZEOF(QUERY ( impl_func <* QUERY ( pdr <* USEDIN(SELF.
of_shape.definition,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | (pdr.name =
'implemented function') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT') IN TYPEOF(impl_func.
relating_product_definition)) AND (impl_func.
relating_product_definition.frame_of_reference.name =
'functional design usage')) )) = 1));
WR14: ((NOT (SELF.description = 'printed part template')) OR (
SIZEOF(QUERY ( cs <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'cross section') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(cs.
relating_shape_aspect)) AND (cs.relying_shape_aspect.
description = 'printed part cross section template')) )) <=
1));
WR15: ((NOT (SELF.description = 'printed part template')) OR (
SIZEOF(QUERY ( ad <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'associated definition') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(ad.
related_shape_aspect)) AND (ad.relying_shape_aspect.
description IN ['interface terminal','join terminal']))) ))
>= 2));
WR16: ((NOT (SELF.description = 'trace template')) OR (SIZEOF(

```

```

QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( si <* QUERY ( it <* pdr.used_representation.items
| ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STYLED_ITEM') IN TYPEOF(it)) ) | ((si.name = 'trace style')
AND (SIZEOF(QUERY ( psa <* si.styles | (NOT (SIZEOF(
QUERY ( sty <* psa.styles | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CURVE_STYLE') IN TYPEOF(sty)) )) = 1)) )) = 0)) )) = 1) ))
= 1)) )) = 0));
WR17: ((NOT (SELF.description = 'unsupported passage template')) OR
(SIZEOF(QUERY ( upt <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'unsupported passage technology') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY') IN TYPEOF(upt.relateing_shape_aspect))
AND (upt.relateing_shape_aspect.description =
'default unsupported passage definition')) )) = 1));
WR18: ((NOT (SELF.description = 'via template')) OR (SIZEOF(
QUERY ( vpt <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'via passage technology') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PASSAGE_TECHNOLOGY') IN TYPEOF(vpt.relateing_shape_aspect))
AND (vpt.relateing_shape_aspect.description =
'default via definition')) )) = 1));
WR19: ((NOT (SELF.description = 'via template')) OR (SIZEOF(
QUERY ( am <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'access mechanism') ) | (am.related_shape_aspect.
description = 'via template terminal') )) >= 2));
WR20: (SIZEOF(QUERY ( esr <* USEDIN(SELF.source,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE') | (esr.name
= 'revision') )) = 1);
END_ENTITY; -- part_template_definition

```

```

ENTITY part_text_template_definition
  SUBTYPE OF (part_template_definition);
  WHERE

```

```

  WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```

```

QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEXT_LITERAL') IN TYPEOF(it)) )) = 1) )) = 1)) )) = 0);
WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum font vertical extent')) )) = 1) )) = 1)) )) = 0);
WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
QUERY ( it <* pdr.used_representation.items | ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum font horizontal extent')) )) = 1) )) = 1)) )) = 0);
END_ENTITY; -- part_text_template_definition

```

```

ENTITY part_tooling_feature
  SUPERTYPE OF (fiducial_part_feature)
  SUBTYPE OF (shape_aspect);
END_ENTITY; -- part_tooling_feature

```

```

ENTITY passage_technology
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1 : (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION') IN TYPEOF(SELF.of_shape.definition));
    WR2 : (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MATERIAL_DESIGNATION.DEFINITIONS')) <= 1);
    WR3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(

```

```

QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') )) = 1)) )) = 0);
WR4 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT ((1 <=
SIZEOF(pcr.used_representation.items)) AND (SIZEOF(pcr.
used_representation.items) <= 8))) )) = 0)) )) = 0);
WR5 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | (NOT (
it.name IN ['maximum aspect ratio','plated passage',
'maximum allowed component terminal extent',
'minimum allowed component terminal extent',
'maximum as finished deposition thickness',
'minimum as finished deposition thickness',
'maximum as finished passage extent',
'minimum as finished passage extent']))) )) = 0)) )) = 0)) ))
= 0);
WR6 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum aspect ratio') AND (it\measure_with_unit.
value_component > 1)) )) <= 1)) )) = 0)) )) = 0);
WR7 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DESRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
name = 'plated passage') AND (it.description IN ['true',
'false']))) )) = 1)) )) = 0)) )) = 0);
WR8 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum as finished passage extent')) )) <= 1)) )) = 0)) ))
= 0);
WR9 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum as finished deposition thickness')) )) <= 1)) ))
= 0)) )) = 0);
WR10: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
        = 'minimum as finished deposition thickness')) )) <= 1)) ))
        = 0)) )) = 0);
WR11: ((NOT (SELF\shape_aspect.description IN [
        'default component termination passage definition',
        'default via definition'])) OR (SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) = 1));
WR12: ((NOT (SELF\shape_aspect.description IN [
        'default component termination passage definition',
        'default unsupported passage definition',
        'default via definition'])) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'physical characteristics representation') ) | (NOT (
        SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
        SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
        = 'minimum as finished passage extent')) )) = 1)) )) = 0)) ))
        = 0));
WR13: ((NOT (SELF\shape_aspect.description =
        'default component termination passage definition')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'physical characteristics representation') ) | (NOT (
        SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
        SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
        = 'minimum as finished deposition thickness')) )) = 1)) ))
        = 0)) )) = 0));
WR14: ((NOT (SELF\shape_aspect.description =
        'default component termination passage definition')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum allowed component terminal extent')) )) = 1)) ))
= 0)) )) = 0));
WR15: ((NOT (SELF\shape_aspect.description =
'default component termination passage definition')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum allowed component terminal extent')) )) = 1)) ))
= 0)) )) = 0));
WR16: ((NOT (SELF\shape_aspect.description =
'default via definition')) OR (SIZEOF(QUERY ( pd <* USEDIN(
SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum as finished deposition thickness')) )) = 1)) ))
= 0)) )) = 0));
WR17: (SIZEOF(QUERY ( rpt <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'as finished inter stratum extent') ) | ((rpt.
related_shape_aspect.of_shape\property_definition.
description = 'finished stratum extent') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(rpt.
related_shape_aspect.of_shape\property_definition.
definition)) AND (rpt.related_shape_aspect.of_shape\
property_definition.definition.name =
'inter stratum extent')) )) = 1);
END_ENTITY; -- passage_technology

```

```

ENTITY path
  SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
  SUBTYPE OF (topological_representation_item);
  edge_list : LIST [1:?] OF UNIQUE oriented_edge;
  WHERE
    WR1: path_head_to_tail(SELF);
END_ENTITY; -- path

```

```

ENTITY pcurve
  SUBTYPE OF (curve);
  basis_surface : surface;
  reference_to_curve : definitional_representation;
  WHERE
    WR1: (SIZEOF(reference_to_curve\representation.items) = 1);
    WR2: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CURVE'
      IN TYPEOF(reference_to_curve\representation.items[1]));
    WR3: (reference_to_curve\representation.items[1]\
      geometric_representation_item.dim = 2);
END_ENTITY; -- pcurve

```

```

ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
  WHERE
    WR1: (SELF\geometric_tolerance.name = 'perpendicularity');
END_ENTITY; -- perpendicularity_tolerance

```

```

ENTITY person;
  id : identifier;
  last_name : OPTIONAL label;
  first_name : OPTIONAL label;
  middle_names : OPTIONAL LIST [1:?] OF label;
  prefix_titles : OPTIONAL LIST [1:?] OF label;
  suffix_titles : OPTIONAL LIST [1:?] OF label;

```

```

    UNIQUE
      UR1 : id;
    WHERE
      WR1: (EXISTS(last_name) OR EXISTS(first_name));
END_ENTITY; -- person

```

```

ENTITY person_and_organization;
  the_person      : person;
  the_organization : organization;
END_ENTITY; -- person_and_organization

```

```

ENTITY person_and_organization_assignment
  ABSTRACT SUPERTYPE;
  assigned_person_and_organization : person_and_organization;
  role                            : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment

```

```

ENTITY person_and_organization_role;
  name : label;
END_ENTITY; -- person_and_organization_role

```

```

ENTITY person_assignment
  ABSTRACT SUPERTYPE;
  assigned_person : person;
  role           : person_role;
END_ENTITY; -- person_assignment

```

```

ENTITY person_role;
  name : label;
END_ENTITY; -- person_role

```

```

ENTITY personal_address
  SUBTYPE OF (address);
  people      : SET [1:?] OF person;
  description : text;
END_ENTITY; -- personal_address

```

```

ENTITY physical_connectivity_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE

```

```

WR1: (SIZEOF(QUERY ( at <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'associated terminals') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_TERMINAL') IN TYPEOF(at.related_shape_aspect)) ))
    > 1);
WR2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF.of_shape.definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
    | (pdr.name = 'connectivity allocation') )) <= 1);
END_ENTITY; -- physical_connectivity_definition

```

```

ENTITY physical_connectivity_element
  SUBTYPE OF (shape_aspect_relationship, shape_aspect);
  WHERE
    WR1: ((SELF.relating_shape_aspect.description =
      'topological junction') XOR (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONNECTIVITY_SUB_STRUCTURE'] * TYPEOF(SELF.
      relating_shape_aspect)) = 1));
    WR2: ((SELF.related_shape_aspect.description =
      'topological junction') XOR (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONNECTIVITY_SUB_STRUCTURE'] * TYPEOF(SELF.
      related_shape_aspect)) = 1));
    WR3: (SIZEOF(QUERY ( se <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'structure element') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(se.
      relating_shape_aspect)) )) = 1);
    WR4: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT') IN TYPEOF(SELF.of_shape.definition)) AND (
      SELF.of_shape.definition\product_definition.
      frame_of_reference.name = 'physical design')));
END_ENTITY; -- physical_connectivity_element

```

```

ENTITY physical_network
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1: (SIZEOF(QUERY ( cr <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'connectivity requirement') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL') IN TYPEOF(cr.related_shape_aspect)) ))
>= 2);
WR2: (SIZEOF(QUERY ( nt <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'network topology') ) | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT') IN TYPEOF(nt.related_shape_aspect)) ))
>= 1);
END_ENTITY; -- physical_network

```

ENTITY physical_unit

SUPERTYPE OF (ONEOF (bare_die, package, packaged_part,
assembly_definition, interconnect_definition))

SUBTYPE OF (product_definition);

WHERE

```

WR1 : ((NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS') IN TYPEOF(
SELF))) OR (SIZEOF(QUERY ( docs <* SELF\
product_definition_with_associated_documents.
documentation_ids | (docs.kind.product_data_type =
'CAD filename') )) <= 1));
WR2 : ((SIZEOF(QUERY ( adta <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (adta.role.name
= 'creation date') )) + SIZEOF(QUERY ( ada <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DATE_ASSIGNMENT.ITEMS') | (ada.role.name =
'creation date') ))) = 1);
WR3 : (SIZEOF(USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1);
WR4 : ((SIZEOF(QUERY ( apoa <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa
.role.name = 'creator') )) + SIZEOF(QUERY ( apoa <* USEDIN(
SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (apoa.role.name
= 'creator') ))) >= 1);
WR5 : (SIZEOF(USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1);
WR6 : (SELF.frame_of_reference.name IN ['physical design',
'physical design usage']);
WR7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PROPERTY_DEFINITION.DEFINITION') | (pd.name =
        'unit technology') )) <= 1);
WR8 : ((NOT ((SELF.frame_of_reference.name = 'physical design') AND
        (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1))) OR (
        SIZEOF(QUERY ( du <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
        | ((du.relying_product_definition.frame_of_reference.name
        = 'physical design usage') AND (SIZEOF(QUERY ( prpc <*
        USEDIN(du.related_product_definition.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1)) )) = 1));
WR9 : ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
        AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
        of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sa <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(sa)) )) >= 2)) )) = 0));
WR10: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
        AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
        of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'maximum negative component height') )) <= 1)) )) = 0));
WR11: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
        AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
        of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'assembly module') )) >= 1))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'maximum positive component height') )) <= 1)) )) = 0));
WR12: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'assembly module') )) >= 1))) OR (
SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | ((pdr.name =
'implemented function') AND (pdr.
relating_product_definition.frame_of_reference.name =
'functional design usage')) )) = 1));
WR13: ((NOT ((SELF.frame_of_reference.name = 'physical design') AND
(SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'assembly module') )) >= 1))) OR (
SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | (pdr.
relating_product_definition.description =
'laminate component') )) = 0));
WR14: ((NOT ((SELF.frame_of_reference.name = 'physical design') AND
(SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( du <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
| ((du.relying_product_definition.frame_of_reference.name
= 'physical design usage') AND (SIZEOF(QUERY ( prpc <*
USEDIN(du.related_product_definition.formation.of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)) ))
= 1));
WR15: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\

```

```

product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | ((pdr.name =
'implemented function') AND (pdr.
relating_product_definition.frame_of_reference.name =
'functional design usage')) )) = 1));
WR16: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd1 <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(pd1)) AND (SIZEOF(
QUERY ( cd <* USEDIN(pd1,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CHARACTERIZED_DEFINITION') IN TYPEOF(cd)) AND (SIZEOF(
QUERY ( pd2 <* USEDIN(cd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd2.name =
'located interconnect module thickness') AND (SIZEOF(
USEDIN(pd2,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) > 1)) ))
= 1)) )) = 1)) )) <= 1));
WR17: ((NOT (SELF.frame_of_reference.name = 'physical design')) OR (
SIZEOF(QUERY ( du <* QUERY ( pdr <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' +
'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'design usage') )
| (du.relying_product_definition.frame_of_reference.name
= 'physical design usage') )) = 1));
WR18: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (pd.name =
'interconnect module usage view physical characteristics') ))
<= 1));
WR19: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.

```



```

of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
used_representation.name =
'minimum thickness over metal requirement') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(pdr.
used_representation))) )) <= 1)) )) <= 1));
WR20: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
used_representation.name =
'maximum thickness over metal requirement') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(pdr.
used_representation))) )) <= 1)) )) <= 1));
WR21: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
used_representation.name =
'minimum thickness over dielectric requirement') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(pdr.
used_representation))) )) <= 1)) )) <= 1));

```

```

WR22: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd.name =
'interconnect module usage view physical characteristics')
AND (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
used_representation.name =
'maximum thickness over dielectric requirement') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(pdr.
used_representation)))) )) <= 1)) )) <= 1));
WR23: ((NOT ((SELF.frame_of_reference.name = 'physical design usage')
AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.
of_product,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
product_category.name = 'interconnect module') )) >= 1)))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | ((pd.name = 'restraint')
OR (pd.name = 'tolerance specific restraint')) )) <= 1));
END_ENTITY; -- physical_unit

```

```

ENTITY physical_unit_datum
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1 : (SELF\shape_aspect.description IN ['axis','plane','point','']);
    WR2 : (SELF\shape_aspect.product_definitional = FALSE);
    WR3 : (SELF\shape_aspect.name IN ['single datum','common datum','']);
    WR4 : ((NOT (SELF\shape_aspect.name = 'common datum')) OR (SIZEOF(
      QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
        name = 'datum feature usage in datum system') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DATUM_SYSTEM') IN TYPEOF(sar)))) )) >= 1));
    WR5 : ((NOT (SELF\shape_aspect.name = 'common datum')) OR (SIZEOF(
      QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
        name = 'datum feature usage in common datum') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DATUM_SYSTEM') IN TYPEOF(sar)))) )) >= 2));

```

```

WR6 : ((NOT ((SELF\shape_aspect.description = 'axis') AND (SELF\
    shape_aspect.name IN ['common datum','single datum']))) OR
    (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((NOT (
    sar.name = 'reference axis')) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_REFERENCE_FRAME') IN TYPEOF(sar))) ) = 1));
WR7 : ((NOT ((SELF\shape_aspect.description = 'plane') AND (SELF\
    shape_aspect.name IN ['common datum','single datum']))) OR
    (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((NOT (
    sar.name = 'reference plane')) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_REFERENCE_FRAME') IN TYPEOF(sar))) ) = 1));
WR8 : ((NOT ((SELF\shape_aspect.description = 'point') AND (SELF\
    shape_aspect.name IN ['common datum','single datum']))) OR
    (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((NOT (
    sar.name = 'reference origin')) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_REFERENCE_FRAME') IN TYPEOF(sar))) ) = 1));
WR9 : ((NOT (SELF\shape_aspect.name = 'single datum')) OR (SIZEOF(
    QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
    name = 'datum usage in datum system') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_SYSTEM') IN TYPEOF(sar))) ) >= 1));
WR10: ((NOT (SELF\shape_aspect.name = 'single datum')) OR (SIZEOF(
    QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
    name = 'datum feature usage in single datum') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DATUM_SYSTEM') IN TYPEOF(sar))) ) = 1));
WR11: (NOT ((SELF\shape_aspect.name IN ['']) AND (SELF\shape_aspect.
    description IN [''])));
WR12: ((NOT (SELF\shape_aspect.description = 'plane')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
    'datum direction property') ) <= 2));
END_ENTITY; -- physical_unit_datum

```

```

ENTITY physical_unit_datum_feature
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.' + 'DEFINITION') | (pd.description =
        'datum feature identification') )) = 1);
    WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name IN ['datum feature usage in common datum',
        'datum feature usage in single datum']) )) >= 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'datum feature usage in single datum') )) <= 1);
END_ENTITY; -- physical_unit_datum_feature

ENTITY physical_unit_datum_target
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'datum target usage') )) >= 1);
END_ENTITY; -- physical_unit_datum_target

ENTITY physical_unit_datum_target_set
    SUBTYPE OF (physical_unit_datum_feature);
    WHERE
        WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
            name = 'datum target usage') )) >= 1);
        WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'constituent') )) = 0);
END_ENTITY; -- physical_unit_datum_target_set

ENTITY physical_unit_geometric_tolerance
    SUBTYPE OF (geometric_tolerance, property_definition);
    WHERE
        WR1: (SELF\geometric_tolerance.name = SELF\property_definition.name);
        WR2: (SELF\geometric_tolerance.toleranced_shape_aspect = SELF\
            property_definition.definition);
        WR3: ((NOT (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LINEAR_PROFILE_TOLERANCE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SURFACE_PROFILE_TOLERANCE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE'] * TYPEOF(SELF)) = 0)) OR (SELF\
        geometric_tolerance.name IN ['circularity','cylindricity',
        'flatness','straightness']));
    WR4: ((NOT (SELF\geometric_tolerance.name = 'circularity')) OR (NOT
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF))));
    WR5: ((NOT (SELF\geometric_tolerance.name = 'cylindricity')) OR (
        NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF))));
    WR6: ((NOT (SELF\geometric_tolerance.name = 'flatness')) OR (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF))));
    WR7: (SIZEOF(QUERY ( pugt <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'group geometric tolerance') ) | (pugt.
        relating_shape_aspect.description =
        'simultaneous requirement') )) <= 1);
END_ENTITY; -- physical_unit_geometric_tolerance

```

```

ENTITY placement
    SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
        axis2_placement_3d))
    SUBTYPE OF (geometric_representation_item);
    location : cartesian_point;
END_ENTITY; -- placement

```

```

ENTITY planar_extent
    SUBTYPE OF (geometric_representation_item);
    size_in_x : length_measure;
    size_in_y : length_measure;
END_ENTITY; -- planar_extent

```

```

ENTITY plane
    SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

```

```

ENTITY plane_angle_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PLANE_ANGLE_UNIT'
            IN TYPEOF(SELF\measure_with_unit.unit_component));

```

```
END_ENTITY; -- plane_angle_measure_with_unit
```

```
ENTITY plane_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- plane_angle_unit
```

```
ENTITY plated_cutout_edge_segment
  SUBTYPE OF (cutout_edge_segment, plated_inter_stratum_feature);
  WHERE
    WR1: (SELF.description IN ['plated cutout edge segment']);
    WR2: (SIZEOF(QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'join implementation') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relate_shape_aspect))
      AND (ji.relate_shape_aspect.name = 'inter stratum join')) ))
      <= 1);
    WR3: ((NOT (SELF.description = 'plated cutout edge segment')) OR (
      SIZEOF(QUERY ( cc <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'composed cutout') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PLATED_INTER_STRATUM_FEATURE') IN TYPEOF(cc.
      relate_shape_aspect)) AND (cc.relate_shape_aspect.
      description = 'plated cutout')) )) = 1));
END_ENTITY; -- plated_cutout_edge_segment
```

```
ENTITY plated_inter_stratum_feature
  SUPERTYPE OF (ONEOF (plated_passage, plated_cutout_edge_segment,
    plated_interconnect_module_edge_segment))
  SUBTYPE OF (inter_stratum_feature);
  WHERE
    WR1: ((SELF.description IN ['bonded conductive base blind via',
      'buried via', 'component termination passage',
      'interfacial connection', 'non conductive base blind via',
      'plated conductive base blind via', 'plated cutout',
      'plated interconnect module edge']) OR (SIZEOF([
```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_CUTOUT_EDGE_SEGMENT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_INTERCONNECT_MODULE_EDGE_SEGMENT'] * TYPEOF(SELF)) =
        1));
    WR2: (SIZEOF(QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'join implementation') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relateing_shape_aspect))
        AND (ji.relateing_shape_aspect.name = 'inter stratum join')) ))
        <= 1);
END_ENTITY; -- plated_inter_stratum_feature

```

```

ENTITY plated_interconnect_module_edge_segment
  SUBTYPE OF (interconnect_module_edge_segment,
    plated_inter_stratum_feature);
WHERE
  WR1: (SELF.description IN ['plated interconnect module edge segment']);
  WR2: (SIZEOF(QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'join implementation') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relateing_shape_aspect))
        AND (ji.relateing_shape_aspect.name = 'inter stratum join')) ))
        <= 1);
  WR3: ((NOT (SELF.description =
        'plated interconnect module edge segment')) OR (SIZEOF(
        QUERY ( ce <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'composed edge') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_INTER_STRATUM_FEATURE') IN TYPEOF(ce.
        relateing_shape_aspect)) AND (ce.relateing_shape_aspect.
        description = 'plated interconnect module edge')) )) = 1));
END_ENTITY; -- plated_interconnect_module_edge_segment

```

```

ENTITY plated_passage
  SUBTYPE OF (plated_inter_stratum_feature);
WHERE
  WR1: (SELF.description IN ['bonded conductive base blind via',
        'buried via', 'component termination passage',
        'interfacial connection', 'non conductive base blind via',
        'plated conductive base blind via']);
  WR2: ((NOT (SELF.description = 'bonded conductive base blind via'))
        OR (SIZEOF(QUERY ( fj <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'features join') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_CONCEPT_RELATIONSHIP') IN TYPEOF(fj.
relating_shape_aspect)) AND (fj.relateing_shape_aspect.name =
'stratum feature conductive join')) )) = 1));
WR3: ((NOT (SELF.description = 'component termination passage')) OR
(SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relateing_shape_aspect.
description = 'component termination passage template')) ))
= 1));
WR4: ((NOT (SELF.description = 'component termination passage')) OR
(SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT') IN TYPEOF(ac.related_shape_aspect))
AND (ac.related_shape_aspect.description =
'component termination passage interface terminal')) )) = 1));
WR5: ((NOT (SELF.description = 'component termination passage')) OR
(SIZEOF(QUERY ( ac <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
name = 'associated component') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL') IN TYPEOF(ac.related_shape_aspect))
AND (ac.related_shape_aspect.description =
'component termination passage join terminal')) )) >= 1));
WR6: ((NOT (SELF.description IN ['bonded conductive base blind via',
'buried via', 'interfacial connection',
'non conductive base blind via',
'plated conductive base blind via'])) OR (SIZEOF(
QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relateing_shape_aspect.
description = 'via template')) )) = 1));
END_ENTITY; -- plated_passage

```

```

ENTITY plus_minus_tolerance;
    range          : tolerance_method_definition;
    toleranced_dimension : dimensional_characteristic;

```



```

    UNIQUE
    UR1 : toleranced_dimension;
END_ENTITY; -- plus_minus_tolerance

```

```

ENTITY point
    SUPERTYPE OF (ONEOF (cartesian_point, point_on_curve, point_on_surface,
        point_replica, degenerate_pcurve))
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point

```

```

ENTITY point_on_curve
    SUBTYPE OF (point);
    basis_curve : curve;
    point_parameter : parameter_value;
END_ENTITY; -- point_on_curve

```

```

ENTITY point_on_surface
    SUBTYPE OF (point);
    basis_surface : surface;
    point_parameter_u : parameter_value;
    point_parameter_v : parameter_value;
END_ENTITY; -- point_on_surface

```

```

ENTITY point_replica
    SUBTYPE OF (point);
    parent_pt : point;
    transformation : cartesian_transformation_operator;
    WHERE
        WR1: (transformation.dim = parent_pt.dim);
        WR2: acyclic_point_replica(SELF, parent_pt);
END_ENTITY; -- point_replica

```

```

ENTITY poly_loop
    SUBTYPE OF (loop, geometric_representation_item);
    polygon : LIST [3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- poly_loop

```

```

ENTITY polyline
    SUBTYPE OF (bounded_curve);
    points : LIST [2:?] OF cartesian_point;
END_ENTITY; -- polyline

```

```

ENTITY position_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
  WHERE
    WR1: ((NOT (SELF\geometric_tolerance.name =
      'boundary based position')) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF)));
    WR2: ((NOT (SELF\geometric_tolerance.name =
      'boundary based position')) OR (SIZEOF(QUERY ( gtr <*
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM') IN
          TYPEOF(gtr.related_geometric_tolerance)) AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(gtr.
              related_geometric_tolerance)) AND (gtr.
                related_geometric_tolerance.name IN ['surface profile',
                  'surface profile locating', 'surface profile refinement'])
              AND (gtr.name = 'profile control')))) <= 1)));
    WR3: ((NOT (SELF\geometric_tolerance.name =
      'feature relating position')) OR (SIZEOF(QUERY ( gtr <*
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
        | ((gtr.name = 'position refining control') AND ((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'POSITION_TOLERANCE') IN TYPEOF(gtr.
            relating_geometric_tolerance)) AND (gtr.
              relating_geometric_tolerance.name IN [
                'feature locating position', 'feature relating position']))) )
        = 1)));
    WR4: ((NOT (SELF\geometric_tolerance.name =
      'feature locating position')) OR (SIZEOF(QUERY ( gtr <*
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'POSITION_TOLERANCE') IN TYPEOF(gtr.
            related_geometric_tolerance)) AND (gtr.
              related_geometric_tolerance.name =
                'feature relating position') AND (gtr.name =
                  'position refining control')) ) = 1)));
    WR5: ((NOT (SELF\geometric_tolerance.name =
      'feature relating position')) OR (SIZEOF(QUERY ( gtr <*
        USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITION_TOLERANCE') IN TYPEOF(gtr.
        related_geometric_tolerance)) AND (gtr.
        related_geometric_tolerance.name =
        'feature relating position') AND (gtr.name =
        'position refining control')) )) <= 1));
END_ENTITY; -- position_tolerance

ENTITY positional_boundary
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SELF\shape_aspect.description IN [
      'dimension related positional boundary',
      'profile related positional boundary']);
    WR2: (SELF\shape_aspect.product_definitional = FALSE);
    WR3: ((NOT (SELF\shape_aspect.description =
      'dimension related positional boundary')) OR (SIZEOF(
      QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'POSITIONAL_BOUNDARY_MEMBER') IN TYPEOF(sar.
      related_shape_aspect)) AND (sar.name = 'boundary member')) ))
      >= 1));
    WR4: ((NOT (SELF\shape_aspect.description =
      'dimension related positional boundary')) OR ((SIZEOF(
      QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
      related_shape_aspect.product_definitional = TRUE) AND (sar.
      name = 'constrained feature')) )) + SIZEOF(QUERY ( sar <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(sar.
      related_shape_aspect)) AND
      (sar.name = 'constrained feature')) )))
      = 1));
    WR5: ((NOT (SELF\shape_aspect.description =
      'dimension related positional boundary')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
      'dimension related positional boundary property') )) = 1));
    WR6: ((NOT (SELF\shape_aspect.description =
      'profile related positional boundary')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
      'profile related positional boundary property') )) = 1));

```

```

WR7: ((NOT (SELF\shape_aspect.description =
'profile related positional boundary')) OR (SIZEOF(
QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITIONAL_BOUNDARY_MEMBER') IN TYPEOF(sar.
related_shape_aspect))) AND (sar.name =
'positional boundary and profile boundary member')) )) = 1));
END_ENTITY; -- positional_boundary

```

```

ENTITY positional_boundary_member
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SELF\shape_aspect.product_definitional = FALSE);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
'positional boundary member property') )) = 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.
name = 'boundary member') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITIONAL_BOUNDARY') IN TYPEOF(sar.relater_shape_aspect)))
AND (sar.relater_shape_aspect.description =
'dimension related positional boundary')) )) = 1);
END_ENTITY; -- positional_boundary_member

```

```

ENTITY pre_defined_curve_font
  SUBTYPE OF (pre_defined_item);
END_ENTITY; -- pre_defined_curve_font

```

```

ENTITY pre_defined_item;
  name : label;
END_ENTITY; -- pre_defined_item

```

```

ENTITY precision_qualifier;
  precision_value : INTEGER;
END_ENTITY; -- precision_qualifier

```

```

ENTITY presentation_representation
  SUBTYPE OF (representation);
  WHERE
    WR1: (SELF\representation.context_of_items\

```

```

        geometric_representation_context.coordinate_space_dimension
        = 2);
    WR2: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_REPRESENTATION_CONTEXT'
        IN TYPEOF(SELF\representation.context_of_items));
END_ENTITY; -- presentation_representation

ENTITY presentation_style_assignment;
    styles : SET [1:?] OF presentation_style_select;
    WHERE
        WR1: (SIZEOF(QUERY ( style1 <* SELF.styles | (NOT (SIZEOF(
            QUERY ( style2 <* (SELF.styles - style1) | (NOT ((TYPEOF(
            style1) <> TYPEOF(style2)) OR (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SURFACE_STYLE_USAGE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EXTERNALLY_DEFINED_STYLE'] * TYPEOF(style1)) = 1))) )) = 0)) ))
            = 0);
        WR2: (SIZEOF(QUERY ( style1 <* SELF.styles | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SURFACE_STYLE_USAGE'
            IN TYPEOF(style1)) )) <= 2);
END_ENTITY; -- presentation_style_assignment

ENTITY presentation_style_by_context
    SUBTYPE OF (presentation_style_assignment);
    style_context : style_context_select;
END_ENTITY; -- presentation_style_by_context

ENTITY primary_orientation_feature
    SUBTYPE OF (physical_unit_datum_feature);
    WHERE
        WR1: (SELF\shape_aspect.product_definitional = TRUE);
        WR2: (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'associated body vertical extent') ) | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_BODY_BOTTOM_SURFACE') IN TYPEOF(it.
            relating_shape_aspect)) )) <= 1);
        WR3: (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
            name = 'associated body vertical extent') ) | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE_BODY_TOP_SURFACE') IN TYPEOF(it.
            relating_shape_aspect)) )) <= 1);
END_ENTITY; -- primary_orientation_feature

```

```

ENTITY primary_reference_terminal
  SUBTYPE OF (package_terminal);
END_ENTITY; -- primary_reference_terminal

```

```

ENTITY printed_component
  SUBTYPE OF (component_definition);
  WHERE
    WR1: (SELF.frame_of_reference.name = 'physical occurrence');
    WR2: (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
      product_category.name = 'interconnect module') )) >= 1);
    WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pt_occ <* QUERY ( sa <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.description =
      'part template occurrence') ) | (SIZEOF(QUERY ( it <*
      QUERY ( sar <* USEDIN(pt_occ,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'instantiated template') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
      relating_shape_aspect)) AND (it.relatating_shape_aspect.
      description IN ['printed part template',
      'printed part cross section template',
      'printed connector template']))) )) = 1) )) = 1))) )) = 0);
    WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( sa <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_TERMINAL') IN TYPEOF(sa)) AND (sa.description =
      'printed component join terminal')) )) >= 1)) )) = 0);
    WR5: ((NOT (SELF.description =
      'join 2 physical connectivity definition supporting')) OR (
      SIZEOF(QUERY ( propd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( ri <*
      QUERY ( propdr <* USEDIN(propd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | (propdr.name = 'requirement implementation') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP') IN TYPEOF(ri.
      relating_property_definition.definition)) AND (ri.

```

```

relating_property_definition.definition.name =
'ordered physical connectivity definition') AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(ri.
relating_property_definition.definition.related_shape_aspect))
AND ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(ri.
relating_property_definition.definition.
relating_shape_aspect))) )) = 1) )) = 1));
WR6: ((NOT (SELF.description = 'printed connector component')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( sa <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT') IN TYPEOF(sa)) AND (sa.description
= 'printed connector component interface terminal')) )) >= 1)) ))
= 0));
WR7: ((NOT (SELF.description = 'printed connector component')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pt_occ <* QUERY ( sa <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (sa.description =
'part template occurrence') ) | (SIZEOF(QUERY ( it <*
QUERY ( sar <* USEDIN(pt_occ,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'instantiated template') ) | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PART_TEMPLATE_DEFINITION') IN TYPEOF(it.
relating_shape_aspect)) AND (it.relying_shape_aspect.
description = 'printed connector template')) )) = 1) )) = 1)) ))
= 0));
WR8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(QUERY ( sar <* USEDIN(sa,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE') IN TYPEOF(sar.relying_shape_aspect)) AND
(sar.name = 'stratum feature implementation')) )) >= 1) ))
>= 1) )) >= 1);
END_ENTITY; -- printed_component

```

```

ENTITY printed_connector_template_terminal_relationship
  SUBTYPE OF (shape_aspect, shape_aspect_relationship);
  WHERE
    WR1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.
      name = 'connector') AND (sar.relateing_shape_aspect.
      description = 'printed connector template')) )) = 1);
    WR2: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect)) AND (SELF\
      shape_aspect_relationship.related_shape_aspect.description =
      'interface terminal')));
    WR3: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(SELF\
      shape_aspect_relationship.related_shape_aspect)) AND (SELF\
      shape_aspect_relationship.related_shape_aspect.description =
      'join terminal')));
  END_ENTITY; -- printed_connector_template_terminal_relationship

ENTITY printed_part_cross_section_template_terminal
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SIZEOF(QUERY ( am <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'access mechanism') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(am.
      relating_shape_aspect)) AND (am.relateing_shape_aspect.
      description = 'printed part cross section template')) )) = 1);
  END_ENTITY; -- printed_part_cross_section_template_terminal

ENTITY printed_part_template_connected_terminals_definition
  SUBTYPE OF (shape_aspect);
  UNIQUE
    UR1 : SELF\shape_aspect.name;
  WHERE
    WR1: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name = 'member connected terminal') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(mct.
      related_shape_aspect)) AND (mct.related_shape_aspect.
      description IN ['interface terminal','join terminal'])) ))
      >= 2);
  END_ENTITY; -- printed_part_template_connected_terminals_definition

```



```

ENTITY printed_part_template_terminal
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SELF.description IN ['interface terminal','join terminal']);
    WR2: (SIZEOF(QUERY ( ga <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT.ITEMS') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_TERMINAL_CONNECTION_ZONE_CATEGORY')
      IN TYPEOF(ga.assigned_group)) )) = 1);
    WR3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      related_shape_aspect.description = 'connection zone') )) >=
      1);
    WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
      QUERY ( pdr <* USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) ))
      >= 1)) )) = 0);
    WR5: (SIZEOF(QUERY ( ad <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'associated definition') ) | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PART_TEMPLATE_DEFINITION') IN TYPEOF(ad.
      relating_shape_aspect)) AND (ad.relying_shape_aspect.
      description = 'printed part template')) )) = 1);
    WR6: (SIZEOF(QUERY ( mct <* QUERY ( sar <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
      name = 'member connected terminal') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION') IN
      TYPEOF(mct.relying_shape_aspect)) )) <= 1);
  END_ENTITY; -- printed_part_template_terminal

```

```

ENTITY printed_part_template_terminal_connection_zone_category
  SUBTYPE OF (group);
  WHERE
    WR1: (SELF\group.name IN ['area edge segment','curve edge segment',
      'surface area','surface point']);
  END_ENTITY; -- printed_part_template_terminal_connection_zone_category

```

```

ENTITY probe_access_area

```

```

SUBTYPE OF (component_shape_aspect);
WHERE
  WR1: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION') IN TYPEOF(SELF.of_shape.definition))
    AND (SIZEOF(QUERY ( prpc <* USEDIN(SELF.of_shape.definition.
    formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
    product_category.name = 'interconnect module') )) >= 1)));
  WR2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    related_shape_aspect.description = 'connection zone') )) = 1);
  WR3: (SIZEOF(QUERY ( pli <* QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
    name = 'probed layout item') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE') IN TYPEOF(pli.related_shape_aspect)) )) =
    1);
  WR4: ((NOT (SELF\shape_aspect.description =
    'internal probe access area')) OR (SIZEOF(QUERY ( i <*
    QUERY ( sar <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'implementation') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_FEATURE') IN TYPEOF(i.relating_shape_aspect)) )) =
    1));
END_ENTITY; -- probe_access_area

```

```

ENTITY product;
  id                : identifier;
  name              : label;
  description       : text;
  frame_of_reference : SET [1:?] OF product_context;
  UNIQUE
  UR1 : id;
END_ENTITY; -- product

```

```

ENTITY product_category;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- product_category

```

```

ENTITY product_category_relationship;
  name      : label;
  description : text;

```

```

        category      : product_category;
        sub_category  : product_category;
    WHERE
        WR1: acyclic_product_category_relationship(SELF, [SELF.sub_category]);
END_ENTITY; -- product_category_relationship

```

```

ENTITY product_concept;
    id          : identifier;
    name        : label;
    description  : text;
    market_context : product_concept_context;
    UNIQUE
        UR1 : id;
END_ENTITY; -- product_concept

```

```

ENTITY product_concept_context
    SUBTYPE OF (application_context_element);
    market_segment_type : label;
END_ENTITY; -- product_concept_context

```

```

ENTITY product_context
    SUBTYPE OF (application_context_element);
    discipline_type : label;
END_ENTITY; -- product_context

```

```

ENTITY product_definition;
    id          : identifier;
    description  : text;
    formation    : product_definition_formation;
    frame_of_reference : product_definition_context;
END_ENTITY; -- product_definition

```

```

ENTITY product_definition_context
    SUBTYPE OF (application_context_element);
    life_cycle_stage : label;
END_ENTITY; -- product_definition_context

```

```

ENTITY product_definition_effectivity
    SUBTYPE OF (effectivity);
    usage : product_definition_relationship;
    UNIQUE
        UR1 : usage, SELF\effectivity.id;
END_ENTITY; -- product_definition_effectivity

```

```

ENTITY product_definition_formation;
    id          : identifier;
    description  : text;
    of_product   : product;
    UNIQUE
        UR1 : id, of_product;
END_ENTITY; -- product_definition_formation

```

```

ENTITY product_definition_formation_relationship;
    id          : identifier;
    name        : label;
    description  : text;
    relating_product_definition_formation : product_definition_formation;
    related_product_definition_formation : product_definition_formation;
END_ENTITY; -- product_definition_formation_relationship

```

```

ENTITY product_definition_formation_with_specified_source
    SUBTYPE OF (product_definition_formation);
    make_or_buy : source;
END_ENTITY; -- product_definition_formation_with_specified_source

```

```

ENTITY product_definition_relationship;
    id          : identifier;
    name        : label;
    description  : text;
    relating_product_definition : product_definition;
    related_product_definition : product_definition;
END_ENTITY; -- product_definition_relationship

```

```

ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
    UNIQUE
        UR1 : SELF\property_definition.definition;
    WHERE
        WR1: (NOT (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_DEFINITION'
            IN TYPEOF(SELF\property_definition.definition)));
END_ENTITY; -- product_definition_shape

```

```

ENTITY product_definition_usage
    SUPERTYPE OF (ONEOF (make_from_usage_option, assembly_component_usage))
    SUBTYPE OF (product_definition_relationship);
    UNIQUE

```

```

    UR1 : SELF\product_definition_relationship.id,
          SELF\product_definition_relationship.relater_product_definition,
          SELF\product_definition_relationship.related_product_definition;
WHERE
    WR1: acyclic_product_definition_relationship(SELF, [SELF\
          product_definition_relationship.related_product_definition],
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_DEFINITION_RELATIONSHIP.'
          + 'RELATED_PRODUCT_DEFINITION');
END_ENTITY; -- product_definition_usage

```

```

ENTITY product_definition_with_associated_documents
    SUBTYPE OF (product_definition);
    documentation_ids : SET [1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents

```

```

ENTITY product_material_composition_relationship
    SUBTYPE OF (product_definition_relationship);
    class : label;
    constituent_amount : SET [1:?] OF measure_with_unit;
    composition_basis : label;
    determination_method : text;
END_ENTITY; -- product_material_composition_relationship

```

```

ENTITY product_related_characterized_product_category
    SUBTYPE OF (characterized_product_category,
    product_related_product_category);
WHERE
    WR1: (SIZEOF(SELF\product_related_product_category.products) = 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') | ((pd.description =
          'product category parameter') AND (SIZEOF(QUERY ( pdr <*
          USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF([
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'MODEL_PARAMETER',
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'MEASURE_REPRESENTATION_ITEM',
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'DESCRIPTIVE_REPRESENTATION_ITEM'] * TYPEOF(pdr.
          used_representation)) = 1) )) = 1)) )) = 1);
    WR3: (SIZEOF(QUERY ( pcr <* USEDIN(SELF,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY') | ((pcr.
          description = 'product value assignment') AND ((
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_RELATED_PRODUCT_CATEGORY') IN TYPEOF(pcr.category))) ))
    = 1);
END_ENTITY; -- product_related_characterized_product_category

```

```

ENTITY product_related_product_category
    SUBTYPE OF (product_category);
    products : SET [1:?] OF product;
END_ENTITY; -- product_related_product_category

```

```

ENTITY projected_zone_definition
    SUBTYPE OF (tolerance_zone_definition);
    projection_end : shape_aspect;
    projected_length : measure_with_unit;
    WHERE
        WR1: (projected_length.value_component > 0);
END_ENTITY; -- projected_zone_definition

```

```

ENTITY promissory_usage_occurrence
    SUBTYPE OF (assembly_component_usage);
END_ENTITY; -- promissory_usage_occurrence

```

```

ENTITY property_definition;
    name : label;
    description : text;
    definition : characterized_definition;
END_ENTITY; -- property_definition

```

```

ENTITY property_definition_relationship;
    name : label;
    description : text;
    relating_property_definition : property_definition;
    related_property_definition : property_definition;
END_ENTITY; -- property_definition_relationship

```

```

ENTITY property_definition_representation;
    definition : property_definition;
    used_representation : representation;
END_ENTITY; -- property_definition_representation

```

```

ENTITY qualitative_uncertainty
    SUBTYPE OF (uncertainty_qualifier);
    uncertainty_value : text;
END_ENTITY; -- qualitative_uncertainty

```

```

ENTITY quantified_assembly_component_usage
  SUBTYPE OF (assembly_component_usage);
    quantity : measure_with_unit;
END_ENTITY; -- quantified_assembly_component_usage

```

```

ENTITY quasi_uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- quasi_uniform_curve

```

```

ENTITY quasi_uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

```

```

ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RATIO_UNIT'
      IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- ratio_measure_with_unit

```

```

ENTITY ratio_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- ratio_unit

```

```

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
    weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
      list_to_array(weights_data,0,
        upper_index_on_control_points);
  WHERE
    WR1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.

```

```

        control_points_list));
    WR2: curve_weights_positive(SELf);
END_ENTITY; -- rational_b_spline_curve

```

```

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
  weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
      make_array_of_array(weights_data,0,u_upper,0,v_upper);
  WHERE
    WR1: ((SIZEOF(weights_data) = SIZEOF(SELf\b_spline_surface.
      control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(
      SELf\b_spline_surface.control_points_list[1])));
    WR2: surface_weights_positive(SELf);
END_ENTITY; -- rational_b_spline_surface

```

```

ENTITY rectangular_composite_surface
  SUBTYPE OF (bounded_surface);
  segments : LIST [1:?] OF LIST [1:?] OF surface_patch;
  DERIVE
    n_u : INTEGER := SIZEOF(segments);
    n_v : INTEGER := SIZEOF(segments[1]);
  WHERE
    WR1: ([ ] = QUERY ( s <* segments | (n_v <> SIZEOF(s)) ));
    WR2: constraints_rectangular_composite_surface(SELf);
END_ENTITY; -- rectangular_composite_surface

```

```

ENTITY rectangular_trimmed_surface
  SUBTYPE OF (bounded_surface);
  basis_surface : surface;
  u1             : parameter_value;
  u2             : parameter_value;
  v1             : parameter_value;
  v2             : parameter_value;
  usense         : BOOLEAN;
  vsense         : BOOLEAN;
  WHERE
    WR1: (u1 <> u2);
    WR2: (v1 <> v2);
    WR3: (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ELEMENTARY_SURFACE'
      IN TYPEOF(basis_surface)) AND (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PLANE'
      IN TYPEOF(basis_surface)))) OR (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

        'SURFACE_OF_REVOLUTION'
            IN TYPEOF(basis_surface)) OR (usense = (u2 > u1)));
WR4: ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SPHERICAL_SURFACE'
        IN TYPEOF(basis_surface)) OR (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TOROIDAL_SURFACE'
        IN TYPEOF(basis_surface)) OR (vsense = (v2 > v1)));
END_ENTITY; -- rectangular_trimmed_surface

```

```

ENTITY reparametrised_composite_curve_segment
    SUBTYPE OF (composite_curve_segment);
    param_length : parameter_value;
    WHERE
        WR1: (param_length > 0);
END_ENTITY; -- reparametrised_composite_curve_segment

```

```

ENTITY representation;
    name          : label;
    items         : SET [1:?] OF representation_item;
    context_of_items : representation_context;
END_ENTITY; -- representation

```

```

ENTITY representation_context;
    context_identifier : identifier;
    context_type       : text;
    INVERSE
        representations_in_context : SET [1:?] OF representation FOR
                                         context_of_items;
END_ENTITY; -- representation_context

```

```

ENTITY representation_item;
    name : label;
    WHERE
        WR1: (SIZEOF(using_representations(SELF)) > 0);
END_ENTITY; -- representation_item

```

```

ENTITY representation_map;
    mapping_origin : representation_item;
    mapped_representation : representation;
    INVERSE
        map_usage : SET [1:?] OF mapped_item FOR mapping_source;
    WHERE
        WR1: item_in_context(SELF.mapping_origin, SELF.mapped_representation.

```

```

        context_of_items);
END_ENTITY; -- representation_map

```

```

ENTITY representation_relationship;
    name      : label;
    description : text;
    rep_1     : representation;
    rep_2     : representation;
END_ENTITY; -- representation_relationship

```

```

ENTITY representation_relationship_with_transformation
    SUBTYPE OF (representation_relationship);
    transformation_operator : transformation;
    WHERE
        WR1: (SELF\representation_relationship.rep_1.context_of_items :<>:
            SELF\representation_relationship.rep_2.context_of_items);
END_ENTITY; -- representation_relationship_with_transformation

```

```

ENTITY requirement_allocation_group
    SUBTYPE OF (group, property_definition_relationship);
    WHERE
        WR1: (SIZEOF(QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_FORMATION',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CONFIGURATION_ITEM'] * TYPEOF(aga.items)) = 1) )) = 1);
        WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REQUIREMENTS_PROPERTY') IN TYPEOF(SELF\
            property_definition_relationship.
            relating_property_definition));
        WR3: ((NOT (SELF\property_definition_relationship.name =
            'derived from operation')) OR (SELF\
            property_definition_relationship.
            relating_property_definition :=: SELF\
            property_definition_relationship.related_property_definition));
END_ENTITY; -- requirement_allocation_group

```

```

ENTITY requirements_property
    SUBTYPE OF (property_definition);
    UNIQUE
        UR1 : SELF\property_definition.name;

```

WHERE

```

WR1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | (pdr.name = 'requirements property composition') )) <= 1);
WR2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( it <* pdr.used_representation.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
    name = 'requirements description')) )) = 1) )) <= 1);
WR3: (SIZEOF(QUERY ( dr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION') IN TYPEOF(dr.assigned_document)) )) >= 1);
WR4: ((NOT (SELF\property_definition.description IN ['constraint',
    'part based constraint'])) OR (SIZEOF(QUERY ( dc <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | (pdr.name = 'design constraint') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_DEFINITION') IN TYPEOF(dc.relatng_property_definition.
    definition)) )) = 1));
WR5: ((NOT (SELF\property_definition.description =
    'part based constraint')) OR (SIZEOF(QUERY ( cp <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | (pdr.name = 'constraining part') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(cp.
    relating_property_definition.definition)) AND (cp.
    relating_property_definition.definition.name =
    'constraining part') AND (cp.relatng_property_definition.
    definition.related_product_definition.frame_of_reference.
    name = 'design requirement')) )) = 1));
WR6: ((NOT (SELF\property_definition.description =
    'interface requirement')) OR (SIZEOF(QUERY ( itnha <*
    QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | (pdr.name = 'interface to next higher assembly') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION') IN TYPEOF(itnha.
    relating_property_definition.definition)) AND (itnha.
    relating_property_definition.definition.frame_of_reference.
    name = 'design requirement') AND (SIZEOF(QUERY ( hai <*
    QUERY ( pdr <* USEDIN(itnha.relatng_property_definition.
    definition,
```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
        | (pdr.name = 'higher assembly interface') ) | (SIZEOF(
        QUERY ( pdr <* USEDIN(hai,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SPECIFIED_HIGHER_USAGE_OCCURRENCE') IN TYPEOF(pdr)) )) = 1) ))
        = 1)) )) = 1));
END_ENTITY; -- requirements_property

```

```

ENTITY requirements_property_group
  SUBTYPE OF (requirements_property, group);
  WHERE
    WR1: (SIZEOF(QUERY ( rpc <* QUERY ( pdr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
    | (pdr.name = 'requirements property composition') ) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY') IN TYPEOF(rpc.
    related_property_definition)) )) >= 1);
END_ENTITY; -- requirements_property_group

```

```

ENTITY revolved_area_solid
  SUBTYPE OF (swept_area_solid);
  axis : axis1_placement;
  angle : plane_angle_measure;
  DERIVE
    axis_line : line := dummy_gri || curve() || line(axis.location,
    dummy_gri || vector(axis.z,1));
END_ENTITY; -- revolved_area_solid

```

```

ENTITY revolved_face_solid
  SUBTYPE OF (swept_face_solid);
  axis : axis1_placement;
  angle : plane_angle_measure;
  DERIVE
    axis_line : line := dummy_gri || curve() || line(axis.location,
    dummy_gri || vector(axis.z,1));
END_ENTITY; -- revolved_face_solid

```

```

ENTITY right_angular_wedge
  SUBTYPE OF (geometric_representation_item);
  position : axis2_placement_3d;
  x : positive_length_measure;
  y : positive_length_measure;

```

```

        z          : positive_length_measure;
        ltx        : length_measure;
    WHERE
        WR1: ((0 <= ltx) AND (ltx < x));
END_ENTITY; -- right_angular_wedge

```

```

ENTITY right_circular_cone
    SUBTYPE OF (geometric_representation_item);
    position      : axis1_placement;
    height        : positive_length_measure;
    radius        : length_measure;
    semi_angle    : plane_angle_measure;
    WHERE
        WR1: (radius >= 0);
END_ENTITY; -- right_circular_cone

```

```

ENTITY right_circular_cylinder
    SUBTYPE OF (geometric_representation_item);
    position      : axis1_placement;
    height        : positive_length_measure;
    radius        : positive_length_measure;
END_ENTITY; -- right_circular_cylinder

```

```

ENTITY routed_printed_component
    SUBTYPE OF (printed_component);
    WHERE
        WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
            USEDIN(pd,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(QUERY ( sar <* USEDIN(sa,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PART_TEMPLATE_DEFINITION') IN TYPEOF(sar.
            relating_shape_aspect))) AND (sar.relatating_shape_aspect.
            description = 'printed part cross section template') AND (
            sar.name = 'instantiated template')) ) = 1) ) = 1) ) = 1));
        WR2: (SIZEOF(QUERY ( csa <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPONENT_SHAPE_ASPECT') IN TYPEOF(csa)) AND (SIZEOF(
            QUERY ( sar <* USEDIN(csa,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((
            SIZEOF(QUERY ( pdr <* USEDIN(sar,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(pdr)) AND (pdr.
        name = 'inter stratum extent')))) = 1) AND (sar.name =
        'component extent')))) = 1)) = 1);
END_ENTITY; -- routed_printed_component

```

ENTITY rule_action

SUBTYPE OF (action);

WHERE

```

    WR1: (SIZEOF(QUERY ( aaa <* QUERY ( aa <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_ACTION_ASSIGNMENT') IN TYPEOF(aa)) ) | (SIZEOF(
        QUERY ( it <* aaa.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_DEFINITION') IN TYPEOF(it)) )) = 1) )) = 1);
    WR2: ((SIZEOF(QUERY ( adta <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (adta.role.name
        = 'participant date and time')) + SIZEOF(QUERY ( ada <*
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | (ada.role.name =
        'participant date')))) = 1);
    WR3: ((NOT (SELF.name = 'rule justification')) OR (SIZEOF(
        QUERY ( ja <* QUERY ( ar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATED_ACTION') | (ar.name =
        'justified action')) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_ACTION') IN TYPEOF(ja.relatng_action)) )) = 1));
    WR4: ((NOT (SELF.name = 'rule modification')) OR (SIZEOF(
        QUERY ( mr <* QUERY ( ar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATED_ACTION') | (ar.name =
        'modification rationale')) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_ACTION') IN TYPEOF(mr.relatng_action)) AND (mr.
        relating_action.name = 'rule change request')))) = 1));
    WR5: ((NOT (SELF.name = 'rule replacement from')) OR (SIZEOF(
        QUERY ( rrfa <* QUERY ( aa <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_REPLACEMENT_FROM_ASSIGNMENT') IN TYPEOF(aa)) ) | (
        SIZEOF(QUERY ( it <* rrfa.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_DEFINITION') IN TYPEOF(it)) )) = 1) )) = 1));

```

```

WR6: ((NOT (SELF.name = 'rule replacement to')) OR (SIZEOF(
    QUERY ( rrta <* QUERY ( aa <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_REPLACEMENT_TO_ASSIGNMENT') IN TYPEOF(aa)) ) | (
    SIZEOF(QUERY ( it <* rrta.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_DEFINITION') IN TYPEOF(it)) )) = 1) )) = 1));
END_ENTITY; -- rule_action

```

```

ENTITY rule_complex_clause
    SUBTYPE OF (representation);
    UNIQUE
        UR1 : SELF\representation.name;
    WHERE
        WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION_RELATIONSHIP.REP_2') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'RULE_SIMPLE_CLAUSE') IN TYPEOF(rr.rep_1)) )) >= 1);
        WR2: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- rule_complex_clause

```

```

ENTITY rule_conclusion_definition
    SUBTYPE OF (representation);
    WHERE
        WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION_RELATIONSHIP.REP_2') | (rr\
            representation_relationship.rep_1\representation.description
            = 'rule definition') )) = 1);
        WR2: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION_RELATIONSHIP.REP_2') | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'RULE_SIMPLE_CLAUSE') IN TYPEOF(rr\
            representation_relationship.rep_1)) )) >= 1);
        WR3: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- rule_conclusion_definition

```

```

ENTITY rule_definition
    SUBTYPE OF (characterized_object, externally_defined_item);
    UNIQUE
        UR1 : SELF\characterized_object.name, SELF\externally_defined_item.item_id;
    WHERE
        WR1: ((SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'APPLIED_DOCUMENT_REFERENCE.ITEMS')) + SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')) <= 1);
WR2: (SIZEOF(QUERY ( aaa <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_ACTION_ASSIGNMENT.ITEMS') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_ACTION') IN TYPEOF(aaa.assigned_action)) )) >= 1);
WR3: (SIZEOF(QUERY ( esr <* USEDIN(SELF.source,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNAL_SOURCE_RELATIONSHIP.RELATING_SOURCE') | (esr.name
        = 'revision') )) = 1);
WR4: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_CONCLUSION_DEFINITION') IN TYPEOF(rr\
        representation_relationship.rep_2)) )) = 1);
WR5: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_PREMISE_DEFINITION') IN TYPEOF(rr\
        representation_relationship.rep_2)) )) = 1);
END_ENTITY; -- rule_definition

```

ENTITY rule_function_definition

SUBTYPE OF (representation);

WHERE

```

WR1: (SIZEOF(QUERY ( it <* SELF.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MODEL_PARAMETER') IN TYPEOF(it)) )) >= 1);
WR2: (SIZEOF(QUERY ( adf <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((adf\
        document_reference.assigned_document.kind.product_data_type
        = 'reference document') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EE_SPECIFICATION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DOCUMENT'] * TYPEOF(adf\document_reference.
        assigned_document)) >= 1)) )) = 1);
WR3: (SIZEOF(QUERY ( adf <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((adf\
        document_reference.assigned_document.kind.product_data_type
        = 'source code') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EE_SPECIFICATION') IN TYPEOF(adf\document_reference.
        assigned_document)))) )) = 1);

```

END_ENTITY; -- rule_function_definition


```

ENTITY rule_premise_definition
  SUBTYPE OF (representation);
  WHERE
    WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_2') | (rr\
      representation_relationship.rep_1\representation.description
      = 'rule definition') )) = 1);
    WR2: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_2') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_COMPLEX_CLAUSE') IN TYPEOF(rr\
      representation_relationship.rep_1)) )) >= 1);
    WR3: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- rule_premise_definition

```

```

ENTITY rule_replacement_from_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF replacement_from_item;
  WHERE
    WR1: (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ACTION_RELATIONSHIP.' + 'RELATING_ACTION')) = 1);
END_ENTITY; -- rule_replacement_from_assignment

```

```

ENTITY rule_replacement_from_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF replacement_from_item;
END_ENTITY; -- rule_replacement_from_request_assignment

```

```

ENTITY rule_replacement_to_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF replacement_to_item;
END_ENTITY; -- rule_replacement_to_assignment

```

```

ENTITY rule_replacement_to_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF replacement_to_item;
  WHERE
    WR1: (SIZEOF(QUERY ( rrfa <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_REPLACEMENT_FROM_ASSIGNMENT') IN TYPEOF(rrfa)) )) >= 1);
END_ENTITY; -- rule_replacement_to_request_assignment

```

```

ENTITY rule_set
  SUBTYPE OF (group);
  UNIQUE
    UR1 : SELF\group.name;
  WHERE
    WR1: ((SIZEOF(QUERY ( aga <* QUERY ( ga <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_GROUP_ASSIGNMENT') IN TYPEOF(ga)) ) | (SIZEOF(
      QUERY ( rd <* QUERY ( it <* aga.items | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_DEFINITION') IN TYPEOF(it)) ) | (SIZEOF(QUERY ( pd <*
      USEDIN(rd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
      used_representation.name = 'rule priority') )) >= 1) )) >= 1) ))
      >= 1) )) >= 1) OR (SIZEOF(QUERY ( rsge <* QUERY ( gr <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GROUP_RELATIONSHIP.RELATING_GROUP') | (gr.name =
      'rule set group element') ) | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_SET') IN TYPEOF(rsge.related_group)) )) >= 2));
END_ENTITY; -- rule_set

```

```

ENTITY rule_simple_clause
  SUBTYPE OF (representation);
  UNIQUE
    UR1 : SELF\representation.name;
  WHERE
    WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_1') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PARAMETER_ASSIGNMENT_REPRESENTATION') IN TYPEOF(rr.rep_2)) ))
      >= 1);
    WR2: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REPRESENTATION_RELATIONSHIP.REP_2') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'RULE_FUNCTION_DEFINITION') IN TYPEOF(rr.rep_1)) )) = 1);
    WR3: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- rule_simple_clause

```

```

ENTITY runout_zone_definition

```

```

    SUBTYPE OF (tolerance_zone_definition);
        orientation : runout_zone_orientation;
END_ENTITY; -- runout_zone_definition

```

```

ENTITY runout_zone_orientation;
    angle : measure_with_unit;
END_ENTITY; -- runout_zone_orientation

```

```

ENTITY runout_zone_orientation_reference_direction
    SUBTYPE OF (runout_zone_orientation);
        orientation_defining_relationship : shape_aspect_relationship;
END_ENTITY; -- runout_zone_orientation_reference_direction

```

```

ENTITY seam_curve
    SUBTYPE OF (surface_curve);
    WHERE
        WR1: (SIZEOF(SELF\surface_curve.associated_geometry) = 2);
        WR2: (associated_surface(SELF\surface_curve.associated_geometry[1])
            = associated_surface(SELF\surface_curve.associated_geometry[
                2]));
        WR3: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PCURVE'
            IN TYPEOF(SELF\surface_curve.associated_geometry[1]));
        WR4: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PCURVE'
            IN TYPEOF(SELF\surface_curve.associated_geometry[2]));
END_ENTITY; -- seam_curve

```

```

ENTITY seating_plane
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (SIZEOF(TYPEOF(SELF)) = 2);
END_ENTITY; -- seating_plane

```

```

ENTITY secondary_orientation_feature
    SUBTYPE OF (physical_unit_datum_feature);
    WHERE
        WR1: (SIZEOF(TYPEOF(SELF)) * [
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRIMARY_ORIENTATION_FEATURE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SEATING_PLANE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'VIEWING_PLANE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT_DATUM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT']) = 0);
WR2: (SELF.product_definitional = TRUE);
WR3: (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated body vertical extent') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY_BOTTOM_SURFACE') IN TYPEOF(it.
        relating_shape_aspect)) )) <= 1);
WR4: (SIZEOF(QUERY ( it <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'associated body vertical extent') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY_TOP_SURFACE') IN TYPEOF(it.
        relating_shape_aspect)) )) <= 1);
END_ENTITY; -- secondary_orientation_feature

```

```

ENTITY security_classification;
    name          : label;
    purpose        : text;
    security_level : security_classification_level;
END_ENTITY; -- security_classification

```

```

ENTITY security_classification_assignment
    ABSTRACT SUPERTYPE;
    assigned_security_classification : security_classification;
END_ENTITY; -- security_classification_assignment

```

```

ENTITY security_classification_level;
    name : label;
END_ENTITY; -- security_classification_level

```

```

ENTITY sequential_laminate_passage_based_fabrication_joint
    SUBTYPE OF (shape_aspect);
END_ENTITY; -- sequential_laminate_passage_based_fabrication_joint

```

```

ENTITY serial_numbered_effectivity
    SUBTYPE OF (effectivity);
    effectivity_start_id : identifier;
    effectivity_end_id   : OPTIONAL identifier;
END_ENTITY; -- serial_numbered_effectivity

```

```

ENTITY shape_aspect;
    name          : label;
    description    : text;
    of_shape       : product_definition_shape;
    product_definitional : LOGICAL;
END_ENTITY; -- shape_aspect

```

```

ENTITY shape_aspect_deriving_relationship
    SUBTYPE OF (shape_aspect_relationship);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DERIVED_SHAPE_ASPECT'
            IN TYPEOF(SELF\shape_aspect_relationship.
                relating_shape_aspect));
END_ENTITY; -- shape_aspect_deriving_relationship

```

```

ENTITY shape_aspect_relationship;
    name          : label;
    description    : text;
    relating_shape_aspect : shape_aspect;
    related_shape_aspect  : shape_aspect;
END_ENTITY; -- shape_aspect_relationship

```

```

ENTITY shape_definition_representation
    SUBTYPE OF (property_definition_representation);
    WHERE
        WR1: ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_DEFINITION'
            IN TYPEOF(SELF.definition.definition)) OR (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(SELF.definition)));
        WR2: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_REPRESENTATION'
            IN TYPEOF(SELF.used_representation));
END_ENTITY; -- shape_definition_representation

```

```

ENTITY shape_dimension_representation
    SUBTYPE OF (shape_representation);
    WHERE
        WR1: (SIZEOF(QUERY ( temp <* SELF.items | (NOT (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MEASURE_REPRESENTATION_ITEM'

```

```

        IN TYPEOF(temp))) )) = 0);
WR2: (SIZEOF(SELF.items) <= 2);
WR3: (SIZEOF(QUERY ( pos_mri <* QUERY ( real_mri <* SELF.items | (
    'REAL' IN TYPEOF(real_mri\measure_with_unit.value_component)) )
    | (NOT (pos_mri\measure_with_unit.value_component > 0)) ))
    = 0);
END_ENTITY; -- shape_dimension_representation

```

ENTITY shape_modification

SUBTYPE OF (shape_aspect);

WHERE

```

    WR1: (SIZEOF(QUERY ( dim <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'design intent modification') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(dim.relateing_shape_aspect)) ))
        = 1);
    WR2: ((SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) AND (pd.description
        = 'modification causal')) )) = 1) OR (SIZEOF(QUERY ( mcf <*
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'modification causal feature') ) | ((SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE'] * TYPEOF(mcf.relateing_shape_aspect)) = 1)
        OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM') IN TYPEOF(mcf.relateing_shape_aspect.of_shape.
        definition))) )) = 1));
    WR3: ((NOT (SELF.description IN ['electrical isolation removal',
        'thermal isolation removal'])) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) AND (pd.description
        = 'spacing requirement')) )) = 1));
    WR4: ((NOT (SELF.description IN ['electrical isolation removal',
        'thermal isolation removal'])) OR (SIZEOF(QUERY ( di <*
        QUERY ( sar <* USEDIN(SELF,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
name = 'design intent') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(di.relateing_shape_aspect)) AND
(di.relateing_shape_aspect.description IN [
        'conductive filled area', 'connected filled area'])) ) = 1));
WR5: ((NOT (SELF.description = 'thermal isolation removal')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) AND (pd.description
        = 'angular orientation requirement')) ) = 1));
WR6: ((NOT (SELF.description = 'thermal isolation removal')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) AND (pd.description
        = 'effective current capacity requirement')) ) = 1));
END_ENTITY; -- shape_modification

```

```

ENTITY shape_representation
  SUBTYPE OF (representation);
END_ENTITY; -- shape_representation

```

```

ENTITY shape_representation_relationship
  SUBTYPE OF (representation_relationship);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_REPRESENTATION'
      IN (TYPEOF(SELF\representation_relationship.rep_1) + TYPEOF(
        SELF\representation_relationship.rep_2)));
END_ENTITY; -- shape_representation_relationship

```

```

ENTITY shell_based_2d_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1 : (SIZEOF(QUERY ( it <* SELF\representation.items | (NOT (
      SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D'] * TYPEOF(it)) = 1)) ) = 0));

```

```

WR2 : (SIZEOF(QUERY ( it <* SELF\representation.items | (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'] * TYPEOF(it)) = 1) )) >= 1);
WR3 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
        shell_based_wireframe_model.sbwm_boundary | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
            QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
            wire_shell_extent | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
                QUERY ( el <* eloop\path.edge_list | (NOT ((
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'EDGE_CURVE') IN TYPEOF(el.edge_element))) )) = 0)) )) = 0)) ))
        = 0)) )) = 0);
WR4 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
        shell_based_wireframe_model.sbwm_boundary | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
            QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
            wire_shell_extent | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
                QUERY ( pline_el <* QUERY ( el <* eloop\path.edge_list | ((
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'POLYLINE') IN TYPEOF(el.edge_element\edge_curve.
                    edge_geometry)) ) | (NOT (SIZEOF(pline_el.edge_element\
                    edge_curve.edge_geometry\polyline.points) > 2)) )) = 0)) ))
        = 0)) )) = 0)) )) = 0);
WR5 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
        shell_based_wireframe_model.sbwm_boundary | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
            QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
            wire_shell_extent | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
                QUERY ( el <* eloop\path.edge_list | (NOT

```



```

        valid_2d_wireframe_edge_curve(el.edge_element\edge_curve.
        edge_geometry,

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
) ))
    = 0)) )) = 0)) )) = 0)) )) = 0);
WR6 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
    SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
    shell_based_wireframe_model.sbwm_boundary | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
    QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
    wire_shell_extent | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
    QUERY ( el <* eloop\path.edge_list | (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VERTEX_POINT') IN TYPEOF(el.edge_element.edge_start)) AND
    (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'VERTEX_POINT') IN TYPEOF(el.edge_element.edge_end)))) ))
    = 0)) )) = 0)) )) = 0)) )) = 0);
WR7 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
    SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
    shell_based_wireframe_model.sbwm_boundary | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
    QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
    wire_shell_extent | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
    QUERY ( el <* eloop\path.edge_list | (NOT (
    valid_wireframe_vertex_point(el.edge_element.edge_start\
    vertex_point.vertex_geometry) AND
    valid_wireframe_vertex_point(el.edge_element.edge_end\
    vertex_point.vertex_geometry)))) )) = 0)) )) = 0)) )) = 0)) ))
    = 0);
WR8 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
    items | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
    SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
    shell_based_wireframe_model.sbwm_boundary | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
    QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
    wire_shell_extent | ((

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP') IN TYPEOF(wsb)) ) | (NOT (SIZEOF(
QUERY ( con_edges <* QUERY ( el <* eloop\path.edge_list | (
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC') IN TYPEOF(el.edge_element\edge_curve.
edge_geometry))) ) | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_2D') IN TYPEOF(con_edges.edge_element\
edge_curve.edge_geometry\conic.position))) )) = 0)) )) = 0)) ))
= 0)) )) = 0);
WR9 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
shell_based_wireframe_model.sbwm_boundary | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
wire_shell_extent | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP') IN TYPEOF(wsb)) ) | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT') IN TYPEOF(vloop\vertex_loop.loop_vertex))) ))
= 0)) )) = 0)) )) = 0);
WR10: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
SIZEOF(QUERY ( ws <* QUERY ( sb <* sbwm\
shell_based_wireframe_model.sbwm_boundary | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL') IN TYPEOF(sb)) ) | (NOT (SIZEOF(
QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
wire_shell_extent | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP') IN TYPEOF(wsb)) ) | (NOT
valid_wireframe_vertex_point(vloop\vertex_loop.loop_vertex\
vertex_point.vertex_geometry))) )) = 0)) )) = 0)) )) = 0);
WR11: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
items | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
SIZEOF(QUERY ( vs <* QUERY ( sb <* sbwm\
shell_based_wireframe_model.sbwm_boundary | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_SHELL') IN TYPEOF(sb)) ) | (NOT ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT') IN TYPEOF(vs\vertex_shell.
vertex_shell_extent.loop_vertex))) )) = 0)) )) = 0);
WR12: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF\representation.
items | ((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL') IN TYPEOF(it)) ) | (NOT (
        SIZEOF(QUERY ( vs <* QUERY ( sb <* sbwm\
        shell_based_wireframe_model.sbwmm_boundary | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VERTEX_SHELL') IN TYPEOF(sb)) ) | (NOT
        valid_wireframe_vertex_point(vs\vertex_shell.
        vertex_shell_extent.loop_vertex\vertex_point.
        vertex_geometry))) )) = 0)) )) = 0);
    WR13: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF\representation.items
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM') IN TYPEOF(it)) ) | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(
        mi\mapped_item.mapping_source.mapped_representation))) )) =
        0);
    WR14: (SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 2);
END_ENTITY; -- shell_based_2d_wireframe_shape_representation

```

```

ENTITY shell_based_surface_model
  SUBTYPE OF (geometric_representation_item);
  sbmm_boundary : SET [1:?] OF shell;
  WHERE
    WR1: constraints_geometry_shell_based_surface_model(SELF);
END_ENTITY; -- shell_based_surface_model

```

```

ENTITY shell_based_wireframe_model
  SUBTYPE OF (geometric_representation_item);
  sbwm_boundary : SET [1:?] OF shell;
  WHERE
    WR1: constraints_geometry_shell_based_wireframe_model(SELF);
END_ENTITY; -- shell_based_wireframe_model

```

```

ENTITY shell_based_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    WR1 : (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_MODEL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MAPPED_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_3D']
        * TYPEOF(it)) = 1)) )) = 0);
    WR2 : (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'SHELL_BASED_WIREFRAME_MODEL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MAPPED_ITEM']
    * TYPEOF(it)) = 1) )) >= 1);
WR3 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
    QUERY ( sb <* sbwm\shell_based_wireframe_model.
    sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL'
    IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( eloop <*
    QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
    edge_list | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_CURVE'
    IN TYPEOF(el.edge_element))) )) = 0)) )) = 0)) )) = 0)) ))
    = 0);
WR4 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
    QUERY ( sb <* sbwm\shell_based_wireframe_model.
    sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL'
    IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( eloop <*
    QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'EDGE_LOOP'
    IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( pline_el <*
    QUERY ( el <* eloop\path.edge_list | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE'
    IN TYPEOF(el.edge_element\edge_curve.edge_geometry)) ) | (
    NOT (SIZEOF(pline_el.edge_element\edge_curve.edge_geometry\
    polyline.points) > 2)) )) = 0)) )) = 0)) )) = 0)) )) = 0);
WR5 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
    IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
    QUERY ( sb <* sbwm\shell_based_wireframe_model.
    sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL'
    IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( eloop <*
    QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

'EDGE_LOOP'
    IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
    edge_list | (NOT valid_wireframe_edge_curve(el.edge_element
    \edge_curve.edge_geometry))) )) = 0)) )) = 0)) )) = 0)) )) =
    0);
WR6 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL'
        IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( eloop <*
        QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_LOOP'
        IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
        edge_list | (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VERTEX_POINT'
        IN TYPEOF(el.edge_element.edge_start)) AND (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VERTEX_POINT'
        IN TYPEOF(el.edge_element.edge_end)))))) )) = 0)) )) = 0)) ))
        = 0)) )) = 0);
WR7 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL'
        IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( eloop <*
        QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_LOOP'
        IN TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
        edge_list | (NOT (valid_wireframe_vertex_point(el.
        edge_element.edge_start\vertex_point.vertex_geometry) AND
        valid_wireframe_vertex_point(el.edge_element.edge_end\
        vertex_point.vertex_geometry)))) )) = 0)) )) = 0)) )) = 0)) ))
        = 0);
WR8 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL'

```

```

        IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( vloop <*
        QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'
        IN TYPEOF(wsb)) ) | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
        IN TYPEOF(vloop\vertex_loop.loop_vertex))) )) = 0)) )) = 0)) ))
= 0);
WR9 : (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'WIRE_SHELL'
        IN TYPEOF(sb)) ) | (NOT (SIZEOF(QUERY ( vloop <*
        QUERY ( wsb <* ws\wire_shell.wire_shell_extent | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_LOOP'
        IN TYPEOF(wsb)) ) | (NOT valid_wireframe_vertex_point(vloop
        \vertex_loop.loop_vertex\vertex_point.vertex_geometry))) ))
= 0)) )) = 0)) )) = 0);
WR10: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_SHELL'
        IN TYPEOF(sb)) ) | (NOT (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_POINT'
        IN TYPEOF(vs\vertex_shell.vertex_shell_extent.loop_vertex))) ))
= 0)) )) = 0);
WR11: (SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_WIREFRAME_MODEL'
        IN TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <*
        QUERY ( sb <* sbwm\shell_based_wireframe_model.
        sbwm_boundary | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VERTEX_SHELL'
        IN TYPEOF(sb)) ) | (NOT valid_wireframe_vertex_point(vs\
        vertex_shell.vertex_shell_extent.loop_vertex\vertex_point.
        vertex_geometry))) )) = 0)) )) = 0);
WR12: (SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MAPPED_ITEM'
        IN TYPEOF(it)) ) | (NOT ((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation))) )) = 0);
    WR13: (SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3);
END_ENTITY; -- shell_based_wireframe_shape_representation

```

```

ENTITY si_unit
  SUBTYPE OF (named_unit);
  prefix : OPTIONAL si_prefix;
  name : si_unit_name;
  DERIVE
    SELF\named_unit.dimensions : dimensional_exponents :=
        dimensions_for_si_unit(SELF.name);
END_ENTITY; -- si_unit

```

```

ENTITY signal
  SUBTYPE OF (characterized_object, property_definition);
  WHERE
    WR1: (SIZEOF(QUERY ( aca <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SIGNAL_CATEGORY') IN TYPEOF(aca.assigned_group))) )) >= 1);
    WR2: (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNAL_DEFINITION') IN TYPEOF(SELF)) XOR (SIZEOF(
        QUERY ( ada <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (SIZEOF(
        QUERY ( duc <* USEDIN(ada.assigned_document,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc.subject_element =
        'signal_category') )) = 1) )) = 1)));
    WR3: (SIZEOF(USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 1);
END_ENTITY; -- signal

```

```

ENTITY signal_category
  SUBTYPE OF (group, externally_defined_item);
  WHERE
    WR1: (SELF\group.description IN ['signal characteristic category',
        'signal property category']);
END_ENTITY; -- signal_category

```

```

ENTITY solid_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);

```

```

WHERE
  WR1: (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SOLID_ANGLE_UNIT'
    IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- solid_angle_measure_with_unit

ENTITY solid_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- solid_angle_unit

ENTITY solid_curve_font
  SUBTYPE OF (pre_defined_curve_font);
END_ENTITY; -- solid_curve_font

ENTITY solid_model
  SUPERTYPE OF (ONEOF (csg_solid, manifold_solid_brep, swept_face_solid,
    swept_area_solid, solid_replica))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- solid_model

ENTITY solid_replica
  SUBTYPE OF (solid_model);
  parent_solid : solid_model;
  transformation : cartesian_transformation_operator_3d;
  WHERE
    WR1: acyclic_solid_replica(SELF, parent_solid);
END_ENTITY; -- solid_replica

ENTITY specified_higher_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
  upper_usage : assembly_component_usage;
  next_usage : next_assembly_usage_occurrence;
  UNIQUE
    UR1 : upper_usage, next_usage;
  WHERE

```



```

WR1: (SELF :<>: upper_usage);
WR2: (SELF\product_definition_relationship.
      relating_product_definition :=: upper_usage.
      relating_product_definition);
WR3: (SELF\product_definition_relationship.
      related_product_definition :=: next_usage.
      related_product_definition);
WR4: (upper_usage.related_product_definition :=: next_usage.
      relating_product_definition);
WR5: (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROMISSORY_USAGE_OCCURRENCE'
      IN TYPEOF(upper_usage)));
END_ENTITY; -- specified_higher_usage_occurrence

```

```

ENTITY sphere
  SUBTYPE OF (geometric_representation_item);
  radius : positive_length_measure;
  centre : point;
END_ENTITY; -- sphere

```

```

ENTITY spherical_surface
  SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- spherical_surface

```

```

ENTITY standard_uncertainty
  SUPERTYPE OF (expanded_uncertainty)
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY; -- standard_uncertainty

```

```

ENTITY start_request
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF start_request_item;
END_ENTITY; -- start_request

```

```

ENTITY start_work
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF work_item;
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'DIRECTED_ACTION') IN TYPEOF(SELF.assigned_action));
END_ENTITY; -- start_work

```

```

ENTITY statistical_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY; -- statistical_dimensional_location

```

```

ENTITY statistical_dimensional_size
  SUBTYPE OF (dimensional_size_property);
  WHERE
    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DIMENSIONAL_SIZE') IN TYPEOF(pdr.
      related_property_definition)) AND (pdr.name =
      'substitutable dimension')) )) <= 1) )) <= 1);
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
      related_property_definition)) AND (pdr.name =
      'dimensional tolerance statistical control requirement')) ))
      <= 1) )) <= 1);
    WR3: ((NOT (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DIMENSIONAL_SIZE') IN TYPEOF(pdr.
      related_property_definition)) AND (pdr.name =
      'substitutable dimension')) )) <= 1) )) <= 1)) OR (NOT (
      SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
      USEDIN(pd,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DIMENSIONAL_SIZE') IN TYPEOF(pdr.

```

```

        related_property_definition))) AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STATISTICAL_DIMENSIONAL_SIZE') IN TYPEOF(pdr.
        related_property_definition))) AND (pdr.name =
        'substitutable dimension')) )) <= 1) )) <= 1)));
END_ENTITY; -- statistical_dimensional_size

ENTITY statistical_geometric_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
  WHERE
    WR1: (SIZEOF(QUERY ( gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE') IN TYPEOF(gtr.
        relating_geometric_tolerance)) AND (gtr.name =
        'substitutable geometric tolerance')) )) <= 1);
    WR2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
        related_property_definition)) AND (pdr.name =
        'geometric tolerance statistical control requirement')) )) =
        1);
    WR3: (SIZEOF(QUERY ( gtr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STATISTICAL_GEOMETRIC_TOLERANCE') IN TYPEOF(gtr.
        relating_geometric_tolerance)) AND (gtr.name =
        'substitutable geometric tolerance')) )) <= 0);
END_ENTITY; -- statistical_geometric_tolerance

ENTITY stratum
  SUBTYPE OF (product_definition);
  UNIQUE
    UR1 : SELF\product_definition.id;
  WHERE
    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'technology usage') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'STRATUM_TECHNOLOGY') IN TYPEOF(tu.
        relating_property_definition.definition)) )) = 1) )) = 1);
WR2: ((SIZEOF(QUERY ( ada <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (SIZEOF(
        QUERY ( duc <* USEDIN(ada.assigned_document,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc.subject_element =
        'attachment region size') )) = 1) )) <= 1) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'stratum usage') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNAL_DEFINITION') IN TYPEOF(tu.
        relating_property_definition)) )) = 1) )) <= 1));
WR3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )
        | (sr_pdr.used_representation.name =
        '3d bound volume shape') )) <= 1)) )) = 0);
WR4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( sr_pdr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation)) )
        | (sr_pdr.used_representation.name =
        'planar projected shape') )) <= 1)) )) = 0);
WR5: (SIZEOF(QUERY ( acu <* QUERY ( pdr <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(pdr)) ) | (acu.name =
        'interconnect module stratum assembly relationship') )) >= 1);
WR6: ((NOT (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'design layer') )) >= 1)) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'technology usage') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_TECHNOLOGY') IN TYPEOF(tu.
        relating_property_definition.definition)) AND (tu.
        relating_property_definition.definition.description =
        'design layer')) ) = 1) ) = 1));
WR7: ((NOT (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'design layer') ) ) >= 1)) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sa <*
        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAYER_CONNECTION_POINT') IN TYPEOF(sa)) ) ) >= 1) ) >= 1));
WR8: ((NOT (SIZEOF(QUERY ( prpc <* USEDIN(SELF.formation.of_product,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS') | (prpc\
        product_category.name = 'documentation layer') ) ) >= 1)) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( tu <*
        QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
        | (pdr.name = 'technology usage') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_TECHNOLOGY') IN TYPEOF(tu.
        relating_property_definition.definition)) AND (tu.
        relating_property_definition.definition.description =
        'documentation layer')) ) = 1) ) = 1));
END_ENTITY; -- stratum

```

ENTITY stratum_concept_relationship

SUBTYPE OF (shape_aspect, shape_aspect_relationship);

WHERE

```

    WR1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REQUIREMENTS_PROPERTY') IN TYPEOF(pd)) ) ) >= 1);

```

```

    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(USEDIN(pd,

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1) )) =
        1);
WR3: ((NOT (SELF\shape_aspect_relationship.name =
        'dielectric crossover area')) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(SELF.relate_shape_aspect))
        AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(SELF.related_shape_aspect))));
WR4: ((NOT (SELF\shape_aspect_relationship.name =
        'dielectric crossover area')) OR (SIZEOF(QUERY ( rdc <*
        QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
        name = 'resulting dielectric crossover') ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(rdc.relate_shape_aspect)) ))
        = 1));
WR5: ((NOT (SELF\shape_aspect_relationship.name =
        'stratum feature conductive join')) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND') IN TYPEOF(SELF.relate_shape_aspect)) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(SELF.
        relate_shape_aspect)) AND (SELF.relate_shape_aspect.
        description = 'stratum feature template component')));
WR6: ((NOT (SELF\shape_aspect_relationship.name =
        'stratum feature conductive join')) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND') IN TYPEOF(SELF.related_shape_aspect)) OR (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT') IN TYPEOF(SELF.
        related_shape_aspect)) AND (SELF.related_shape_aspect.
        description = 'stratum feature template component')));
WR7: ((NOT (SELF\shape_aspect_relationship.name =
        'stratum feature conductive join')) OR (SIZEOF(
        QUERY ( fj <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
        name = 'features join') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_PASSAGE') IN TYPEOF(fj.related_shape_aspect)) AND (
        fj.related_shape_aspect.description =
        'bonded conductive base blind via')) )) <= 1));
WR8: ((NOT ((SELF\shape_aspect.description =
        'physical network supporting stratum feature conductive join')
        AND (SELF\shape_aspect_relationship.name =
        'stratum feature conductive join')) OR (SIZEOF(
        QUERY ( ji <* QUERY ( sar <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.

```

```

        name = 'join implementation') ) | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relater_shape_aspect))
        AND (ji.relater_shape_aspect.name = 'inter stratum join')) ))
        = 1));
END_ENTITY; -- stratum_concept_relationship

```

```

ENTITY stratum_feature
  SUPERTYPE OF (fiducial_stratum_feature)
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM') IN TYPEOF(SELf.of_shape.definition));
    WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(
    QUERY ( it <* pdr.used_representation.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
    name = 'feature of size') AND (it.description IN ['true',
    'false']))) )) = 1) )) = 1)) )) = 0);
    WR3: ((NOT (SELf.description = 'conductor')) OR (SIZEOF(
    QUERY ( ji <* QUERY ( sar <* USEDIN(SELf,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'join implementation') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relater_shape_aspect))
    AND (ji.relater_shape_aspect.name = 'intra stratum join')) ))
    >= 1));
    WR4: ((NOT (SELf.description = 'connected filled area')) OR (SIZEOF(
    QUERY ( ji <* QUERY ( sar <* USEDIN(SELf,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.
    name = 'join implementation') ) | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'JOIN_SHAPE_ASPECT') IN TYPEOF(ji.relater_shape_aspect))
    AND (ji.relater_shape_aspect.name = 'intra stratum join')) ))
    = 1));
END_ENTITY; -- stratum_feature

```

```

ENTITY stratum_surface
  SUBTYPE OF (shape_aspect);
  WHERE
    WR1: (SELf\shape_aspect.description IN ['primary surface',
    'secondary surface', 'average surface']);
    WR2: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'STRATUM') IN TYPEOF(SELF\shape_aspect.of_shape.definition));
END_ENTITY; -- stratum_surface

```

```

ENTITY stratum_technology
  SUBTYPE OF (characterized_object);
  UNIQUE
    UR1 : SELF\characterized_object.name;
  WHERE
    WR1 : (SIZEOF(USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'MATERIAL_DESIGNATION.DEFINITIONS')) = 1);
    WR2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'stiffness class representation') ))
          <= 1)) )) = 0);
    WR3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name =
              'physical characteristics representation') )) = 1)) )) = 0);
    WR4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( scr <* QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'stratum class representation') )
          | (NOT (SIZEOF(QUERY ( it <* scr.used_representation.items
            | (((
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
                name = 'laminate stiffness class') AND (it.description IN [
                  'fluid like', 'conformal coat', 'stiff laminate']))) )) = 1)) ))
            = 0)) )) = 0);
    WR5 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name =
              'physical characteristics representation') ) | (NOT ((3 <=
                SIZEOF(pcr.used_representation.items)) AND (SIZEOF(pcr.
                  used_representation.items) <= 8)))) )) = 0)) )) = 0);

```



```

WR6 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
    used_representation.name =
    'physical characteristics representation') ) | (NOT (
    SIZEOF(QUERY ( it <* pcr.used_representation.items | (NOT (
    it.name IN ['documentation layer purpose',
    'design layer purpose','design layer position',
    'maximum feature size requirement',
    'minimum finished feature spacing',
    'minimum finished feature size','maximum stratum thickness',
    'minimum stratum thickness']))) )) = 0))) )) = 0))) )) = 0);

WR7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
    used_representation.name =
    'physical characteristics representation') ) | (NOT (
    SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
    SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
    = 'minimum finished feature size')) )) = 1))) )) = 0))) )) =
    0);

WR8 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
    used_representation.name =
    'physical characteristics representation') ) | (NOT (
    SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
    SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
    = 'maximum stratum thickness')) )) = 1))) )) = 0))) )) = 0);

WR9 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
    QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.

```

```

used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum stratum thickness')) )) = 1)) )) = 0)) )) = 0);
WR10: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'maximum feature size requirement')) )) <= 1)) )) = 0)) ))
= 0);
WR11: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MEASURE_REPRESENTATION_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
= 'minimum finished feature spacing')) )) <= 1)) )) = 0)) ))
= 0);
WR12: ((NOT (SELF.description = 'documentation layer')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
used_representation.name =
'physical characteristics representation') ) | (NOT (
SIZEOF(QUERY ( it <* pcr.used_representation.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'layer purpose')) )) = 1)) )) = 0)) )) = 0));
WR13: ((NOT (SELF.description = 'design layer')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'physical characteristics representation') ) | (NOT (
        SIZEOF(QUERY ( it <* pcr.used_representation.items | ((
        SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (it.name
        = 'minimum finished feature spacing')) )) = 1)) )) = 0)) ))
        = 0));
WR14: ((NOT (SELF.description = 'design layer')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'physical characteristics representation') ) | (NOT (
        SIZEOF(QUERY ( it <* pcr.used_representation.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'layer purpose') AND (it.description IN [
        'power or ground','other signal','lands only']))) )) = 1)) ))
        = 0)) )) = 0));
WR15: ((NOT (SELF.description = 'design layer')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pcr <* QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name =
        'physical characteristics representation') ) | (NOT (
        SIZEOF(QUERY ( it <* pcr.used_representation.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(it)) AND (it.
        name = 'layer position') AND (it.description IN ['primary',
        'secondary','internal']))) )) = 1)) )) = 0)) )) = 0));
END_ENTITY; -- stratum_technology

```

ENTITY structured_text_representation_context

```

SUBTYPE OF (representation_context);
WHERE
  WR1: (SIZEOF(QUERY ( rep <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.CONTEXT_OF_ITEMS') | (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRUCTURED_TEXT_REPRESENTATION_ITEM') IN TYPEOF(rep))) )) =
    0);
END_ENTITY; -- structured_text_representation_context

ENTITY structured_text_representation_item
SUBTYPE OF (representation, descriptive_representation_item);
WHERE
  WR1: (SIZEOF(QUERY ( rr <* USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_2') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRUCTURED_TEXT_REPRESENTATION_ITEM') IN TYPEOF(rr.rep_1)) ))
    <= 1);
  WR2: (SIZEOF(USEDIN(SELF,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS')) >= 1);
  WR3: (SIZEOF(QUERY ( it <* SELF\representation.items | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_REPRESENTATION_ITEM') IN TYPEOF(it)) ))
    = 1);
END_ENTITY; -- structured_text_representation_item

ENTITY styled_item
SUBTYPE OF (representation_item);
  styles : SET [1:?] OF presentation_style_assignment;
  item : representation_item;
WHERE
  WR1: ((SIZEOF(SELF.styles) = 1) XOR (SIZEOF(QUERY ( pres_style <*
    SELF.styles | (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRESENTATION_STYLE_BY_CONTEXT') IN TYPEOF(pres_style))) ))
    = 0));
END_ENTITY; -- styled_item

ENTITY supplied_part_relationship
SUBTYPE OF (product_definition_relationship);
WHERE
  WR1: (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE') IN
    TYPEOF(SELF.related_product_definition.formation));
  WR2: (NOT (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'PROPERTY_DEFINITION.DEFINITION') IN ROLESOF(SELF)));
END_ENTITY; -- supplied_part_relationship

```

```

ENTITY surface
  SUPERTYPE OF (ONEOF (elementary_surface, swept_surface, bounded_surface,
    offset_surface, surface_replica))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

ENTITY surface_curve
  SUPERTYPE OF (ONEOF (intersection_curve, seam_curve) ANDOR
    bounded_surface_curve)
  SUBTYPE OF (curve);
  curve_3d          : curve;
  associated_geometry : LIST [1:2] OF pcurve_or_surface;
  master_representation : preferred_surface_curve_representation;
  DERIVE
    basis_surface : SET [1:2] OF surface := get_basis_surface(SELF);
  WHERE
    WR1: (curve_3d.dim = 3);
    WR2: ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PCURVE'
      IN TYPEOF(associated_geometry[1])) OR (master_representation
        <> pcurve_s1));
    WR3: ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PCURVE'
      IN TYPEOF(associated_geometry[2])) OR (master_representation
        <> pcurve_s2));
    WR4: (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PCURVE'
      IN TYPEOF(curve_3d)));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
  extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution
  SUBTYPE OF (swept_surface);
  axis_position : axis1_placement;
  DERIVE
    axis_line : line := dummy_gri || curve() || line(axis_position.
      location, dummy_gri || vector(axis_position.z, 1));
END_ENTITY; -- surface_of_revolution

```

```

ENTITY surface_patch
  SUBTYPE OF (founded_item);
    parent_surface : bounded_surface;
    u_transition    : transition_code;
    v_transition    : transition_code;
    u_sense        : BOOLEAN;
    v_sense        : BOOLEAN;
  INVERSE
    using_surfaces : BAG [1:?] OF rectangular_composite_surface FOR
        segments;
  WHERE
    WR1: (NOT (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CURVE_BOUNDED_SURFACE'
      IN TYPEOF(parent_surface)));
END_ENTITY; -- surface_patch

```

```

ENTITY surface_profile_tolerance
  SUBTYPE OF (physical_unit_geometric_tolerance);
  WHERE
    WR1: ((NOT (SELF\geometric_tolerance.name =
      'surface profile refinement')) OR (SIZEOF(QUERY ( gtr <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATED_GEOMETRIC_TOLERANCE')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(gtr.
      relating_geometric_tolerance)) AND ((gtr.
      relating_geometric_tolerance.name =
      'surface profile locating') OR (gtr.
      relating_geometric_tolerance.name =
      'surface profile refinement')) AND (gtr.name =
      'surface profile refining control')) ) = 1)));
    WR2: ((NOT (SELF\geometric_tolerance.name =
      'surface profile locating')) OR (SIZEOF(QUERY ( gtr <*
      USEDIN(SELF,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
      | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(gtr.
      related_geometric_tolerance)) AND (gtr.
      related_geometric_tolerance.name =
      'surface profile refinement') AND (gtr.name =
      'surface profile refining control')) ) = 1)));
    WR3: ((NOT ((NOT ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM') IN
      TYPEOF(SELF))) AND (SELF\geometric_tolerance.name =
      'surface profile refinement')) OR (SIZEOF(QUERY ( gtr <*

```

```

        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(gtr.
        related_geometric_tolerance)) AND (gtr.
        related_geometric_tolerance.name =
        'surface profile refinement') AND (gtr.name =
        'surface profile refining control')))) = 0));
WR4: ((NOT (SELF\geometric_tolerance.name =
        'surface profile refinement')) OR (SIZEOF(QUERY ( gtr < *
        USEDIN(SELF,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_TOLERANCE_RELATIONSHIP.RELATING_GEOMETRIC_TOLERANCE')
        | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SURFACE_PROFILE_TOLERANCE') IN TYPEOF(gtr.
        related_geometric_tolerance)) AND (gtr.
        related_geometric_tolerance.name =
        'surface profile refinement') AND (gtr.name =
        'surface profile refining control')))) = 1));
WR5: (NOT (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF)));
END_ENTITY; -- surface_profile_tolerance

```

```

ENTITY surface_replica
  SUBTYPE OF (surface);
  parent_surface : surface;
  transformation : cartesian_transformation_operator_3d;
  WHERE
    WR1: acyclic_surface_replica(SELF,parent_surface);
END_ENTITY; -- surface_replica

```

```

ENTITY swept_area_solid
  SUPERTYPE OF (ONEOF (revolved_area_solid,extruded_area_solid))
  SUBTYPE OF (solid_model);
  swept_area : curve_bounded_surface;
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PLANE'
      IN TYPEOF(swept_area.basis_surface));
END_ENTITY; -- swept_area_solid

```

```

ENTITY swept_face_solid
  SUPERTYPE OF (ONEOF (extruded_face_solid,revolved_face_solid))
  SUBTYPE OF (solid_model);

```

```

        swept_face : face_surface;
WHERE
    WR1: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLANE'
        IN TYPEOF(swept_face.face_geometry));
END_ENTITY; -- swept_face_solid

ENTITY swept_surface
    SUPERTYPE OF (ONEOF (surface_of_linear_extrusion, surface_of_revolution))
    SUBTYPE OF (surface);
    swept_curve : curve;
END_ENTITY; -- swept_surface

ENTITY symbol_representation
    SUBTYPE OF (representation);
END_ENTITY; -- symbol_representation

ENTITY symbol_representation_map
    SUBTYPE OF (representation_map);
WHERE
    WR1: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SYMBOL_REPRESENTATION'
        IN TYPEOF(SELF\representation_map.mapped_representation));
    WR2: (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT'
        IN TYPEOF(SELF\representation_map.mapping_origin));
END_ENTITY; -- symbol_representation_map

ENTITY symbol_target
    SUBTYPE OF (geometric_representation_item);
    placement : axis2_placement;
    x_scale   : positive_ratio_measure;
    y_scale   : positive_ratio_measure;
END_ENTITY; -- symbol_target

ENTITY symmetric_shape_aspect
    SUBTYPE OF (shape_aspect);
    INVERSE
        basis_relationships : SET [1:?] OF shape_aspect_relationship FOR
                                relating_shape_aspect;
WHERE
    WR1: (SIZEOF(QUERY ( x <* SELF.basis_relationships | (

```



```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CENTRE_OF_SYMMETRY'
        IN TYPEOF(x.related_shape_aspect)) )) >= 1);
END_ENTITY; -- symmetric_shape_aspect

ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
  WHERE
    WR1: (SELF\geometric_tolerance.name = 'symmetry');
END_ENTITY; -- symmetry_tolerance

ENTITY tangent
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    WR1: (SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1);
END_ENTITY; -- tangent

ENTITY text_font;
  id          : identifier;
  name        : label;
  description : text;
  INVERSE
    glyphs : SET [1:?] OF character_glyph_font_usage FOR font;
END_ENTITY; -- text_font

ENTITY text_font_family;
  id          : identifier;
  name        : label;
  description : text;
  INVERSE
    fonts : SET [1:?] OF text_font_in_family FOR family;
END_ENTITY; -- text_font_family

ENTITY text_font_in_family;
  font   : text_font;
  family : text_font_family;
END_ENTITY; -- text_font_in_family

ENTITY text_literal
  SUBTYPE OF (geometric_representation_item);
  literal      : presentable_text;
  placement    : axis2_placement;
  alignment    : text_alignment;
  path         : text_path;

```

```

        font      : font_select;
END_ENTITY; -- text_literal

```

```

ENTITY text_literal_with_associated_curves
  SUBTYPE OF (text_literal);
  associated_curves : SET [1:?] OF curve;
END_ENTITY; -- text_literal_with_associated_curves

```

```

ENTITY text_literal_with_extent
  SUBTYPE OF (text_literal);
  extent : planar_extent;
END_ENTITY; -- text_literal_with_extent

```

```

ENTITY text_string_representation
  SUBTYPE OF (representation);
  WHERE
    WR1: (SIZEOF(QUERY ( item <* SELF\representation.items | (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TEXT_LITERAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANNOTATION_TEXT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANNOTATION_TEXT_CHARACTER',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DEFINED_CHARACTER_GLYPH',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPOSITE_TEXT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT']
      * TYPEOF(item)) = 0) )) = 0);
    WR2: (SIZEOF(QUERY ( item <* SELF\representation.items | (NOT (
      SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TEXT_LITERAL',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANNOTATION_TEXT',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ANNOTATION_TEXT_CHARACTER',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DEFINED_CHARACTER_GLYPH',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPOSITE_TEXT']
      * TYPEOF(item)) = 0) )) >= 1);
    WR3: (SIZEOF(QUERY ( a2p <* QUERY ( item <* SELF\representation.
      items | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'AXIS2_PLACEMENT'
      IN TYPEOF(item)) ) | (NOT ((SIZEOF(QUERY ( at <*

```

```

        QUERY ( item <* SELF\representation.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANNOTATION_TEXT') IN TYPEOF(item)) ) | (at\mapped_item.
        mapping_target :=: a2p) )) >= 1) OR (SIZEOF(QUERY ( atc <*
        QUERY ( item <* SELF\representation.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANNOTATION_TEXT_CHARACTER') IN TYPEOF(item)) ) | (atc\
        mapped_item.mapping_target :=: a2p) )) >= 1))) )) = 0);
END_ENTITY; -- text_string_representation

```

```

ENTITY text_style;
    name : label;
    character_appearance : character_style_select;
END_ENTITY; -- text_style

```

```

ENTITY text_style_for_defined_font;
    text_colour : colour;
END_ENTITY; -- text_style_for_defined_font

```

```

ENTITY thermal_component
    SUBTYPE OF (component_definition);
END_ENTITY; -- thermal_component

```

```

ENTITY thermal_component_shape_aspect
    SUBTYPE OF (component_shape_aspect);
END_ENTITY; -- thermal_component_shape_aspect

```

```

ENTITY thermal_feature
    SUBTYPE OF (shape_aspect);
END_ENTITY; -- thermal_feature

```

```

ENTITY thermal_network
    SUBTYPE OF (functional_unit);
END_ENTITY; -- thermal_network

```

```

ENTITY thermodynamic_temperature_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'THERMODYNAMIC_TEMPERATURE_UNIT'
            IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- thermodynamic_temperature_measure_with_unit

```

```

ENTITY thermodynamic_temperature_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 1) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- thermodynamic_temperature_unit

```

```

ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    WR1: (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'TIME_UNIT'
      IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- time_measure_with_unit

```

```

ENTITY time_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 1) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- time_unit

```

```

ENTITY tolerance_value;
  lower_bound : measure_with_unit;
  upper_bound : measure_with_unit;
  WHERE
    WR1: (upper_bound.value_component > lower_bound.value_component);
    WR2: (upper_bound.unit_component = lower_bound.unit_component);
END_ENTITY; -- tolerance_value

```

```

ENTITY tolerance_zone
  SUBTYPE OF (shape_aspect);
  defining_tolerance : SET [1:?] OF geometric_tolerance;

```

```

        form                : tolerance_zone_form;
END_ENTITY; -- tolerance_zone

```

```

ENTITY tolerance_zone_boundary

```

```

    SUBTYPE OF (shape_aspect);

```

```

    WHERE

```

```

        WR1: ((NOT (SELF\shape_aspect.description = 'conical')) OR (NOT (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF)) OR ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF))))));

```

```

        WR2: ((NOT (SELF\shape_aspect.description =
            'circular or cylindrical or spherical')) OR (NOT (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF)) OR ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF))))));

```

```

        WR3: (SELF\shape_aspect.product_definitional = FALSE);

```

```

        WR4: (NOT (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF)) AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(
            SELF))));

```

```

        WR5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | ((pd.description =
            'boundary zone definition with specified size') OR (pd.
            description = 'conical tolerance zone boundary')) )) = 1);

```

```

        WR6: (SIZEOF(USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'TOLERANCE_ZONE_DEFINITION.BOUNDARIES')) = 1);

```

```

END_ENTITY; -- tolerance_zone_boundary

```

```

ENTITY tolerance_zone_definition

```

```

    SUPERTYPE OF (ONEOF (projected_zone_definition,runout_zone_definition));

```

```

        zone                : tolerance_zone;

```

```

        boundaries : SET [1:?] OF shape_aspect;

```

```

END_ENTITY; -- tolerance_zone_definition

```

```

ENTITY tolerance_zone_explicit_opposing_boundary_set

```

```

    SUBTYPE OF (tolerance_zone_boundary);

```

```

END_ENTITY; -- tolerance_zone_explicit_opposing_boundary_set

```

```

ENTITY tolerance_zone_form;
    name : label;
END_ENTITY; -- tolerance_zone_form

```

```

ENTITY tolerance_zone_implicit_opposing_boundary_set
    SUBTYPE OF (tolerance_zone_boundary);
END_ENTITY; -- tolerance_zone_implicit_opposing_boundary_set

```

```

ENTITY topological_representation_item
    SUPERTYPE OF (ONEOF (vertex, edge, face_bound, face, vertex_shell,
        wire_shell, connected_edge_set, connected_face_set, loop ANDOR path))
    SUBTYPE OF (representation_item);
END_ENTITY; -- topological_representation_item

```

```

ENTITY toroidal_surface
    SUBTYPE OF (elementary_surface);
    major_radius : positive_length_measure;
    minor_radius : positive_length_measure;
END_ENTITY; -- toroidal_surface

```

```

ENTITY torus
    SUBTYPE OF (geometric_representation_item);
    position      : axis1_placement;
    major_radius  : positive_length_measure;
    minor_radius  : positive_length_measure;
    WHERE
        WR1: (major_radius > minor_radius);
END_ENTITY; -- torus

```

```

ENTITY total_runout_tolerance
    SUBTYPE OF (geometric_tolerance_with_specified_datum_system);
    WHERE
        WR1: (SELF\geometric_tolerance.name = 'total runout');
        WR2: (NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'MODIFIED_GEOMETRIC_TOLERANCE') IN TYPEOF(SELF)));
END_ENTITY; -- total_runout_tolerance

```

```

ENTITY trimmed_curve
    SUBTYPE OF (bounded_curve);
    basis_curve      : curve;
    trim_1           : SET [1:2] OF trimming_select;
    trim_2           : SET [1:2] OF trimming_select;
    sense_agreement  : BOOLEAN;

```

```

        master_representation : trimming_preference;
WHERE
    WR1: ((HIINDEX(trim_1) = 1) OR (TYPEOF(trim_1[1]) <> TYPEOF(trim_1[2])));
    WR2: ((HIINDEX(trim_2) = 1) OR (TYPEOF(trim_2[1]) <> TYPEOF(trim_2[2])));
END_ENTITY; -- trimmed_curve

```

```

ENTITY two_direction_repeat_factor
    SUBTYPE OF (one_direction_repeat_factor);
    second_repeat_factor : vector;
END_ENTITY; -- two_direction_repeat_factor

```

```

ENTITY type_qualifier;
    name : label;
END_ENTITY; -- type_qualifier

```

```

ENTITY uncertainty_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    name : label;
    description : text;
WHERE
    WR1: valid_measure_value(SELF\measure_with_unit.value_component);
END_ENTITY; -- uncertainty_measure_with_unit

```

```

ENTITY uncertainty_qualifier
    SUPERTYPE OF (ONEOF (standard_uncertainty, qualitative_uncertainty));
    measure_name : label;
    description : text;
END_ENTITY; -- uncertainty_qualifier

```

```

ENTITY uniform_curve
    SUBTYPE OF (b_spline_curve);
END_ENTITY; -- uniform_curve

```

```

ENTITY uniform_surface
    SUBTYPE OF (b_spline_surface);
END_ENTITY; -- uniform_surface

```

```

ENTITY vector
    SUBTYPE OF (geometric_representation_item);
    orientation : direction;
    magnitude : length_measure;
WHERE
    WR1: (magnitude >= 0);
END_ENTITY; -- vector

```

```

ENTITY versioned_action_request;
    id          : identifier;
    version      : label;
    purpose      : text;
    description   : text;
END_ENTITY; -- versioned_action_request

```

```

ENTITY vertex
    SUBTYPE OF (topological_representation_item);
END_ENTITY; -- vertex

```

```

ENTITY vertex_loop
    SUBTYPE OF (loop);
    loop_vertex : vertex;
END_ENTITY; -- vertex_loop

```

```

ENTITY vertex_point
    SUBTYPE OF (vertex, geometric_representation_item);
    vertex_geometry : point;
END_ENTITY; -- vertex_point

```

```

ENTITY vertex_shell
    SUBTYPE OF (topological_representation_item);
    vertex_shell_extent : vertex_loop;
END_ENTITY; -- vertex_shell

```

```

ENTITY viewing_plane
    SUBTYPE OF (shape_aspect);
    WHERE
        WR1: (SELF\shape_aspect.description = 'affected plane');
        WR2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.' + 'DEFINITION') | (pd.description =
            'viewing plane property') )) = 1);
        WR3: (NOT SELF\shape_aspect.product_definitional);
END_ENTITY; -- viewing_plane

```

```

ENTITY volume_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        WR1: (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'VOLUME_UNIT'

```



```

        IN TYPEOF(SELF\measure_with_unit.unit_component));
END_ENTITY; -- volume_measure_with_unit

```

```

ENTITY volume_unit
  SUBTYPE OF (named_unit);
  WHERE
    WR1: ((SELF\named_unit.dimensions.length_exponent = 3) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- volume_unit

```

```

ENTITY wire_shell
  SUBTYPE OF (topological_representation_item);
  wire_shell_extent : SET [1:?] OF loop;
  WHERE
    WR1: (NOT mixed_loop_type_set(wire_shell_extent));
END_ENTITY; -- wire_shell

```

```

RULE acu_requires_security_classification FOR (assembly_component_usage,
  applied_security_classification_assignment);

```

```

WHERE
  WR1: (SIZEOF(QUERY ( acu <* assembly_component_usage | (NOT (SIZEOF(
    QUERY ( asca <* applied_security_classification_assignment | (
    acu IN asca.items) )) = 1)) )) = 0);
END_RULE; -- acu_requires_security_classification

```

```

RULE add_design_object_management_relationship_unique_constraint FOR (
  add_design_object_assignment,
  add_design_object_request_assignment);

```

```

LOCAL
  pdr_bag : BAG OF product_definition_relationship := [];
  adoa_bag : BAG OF add_design_object_assignment;
  pd_bag : BAG OF product_definition := [];
  mdo_bag : BAG OF managed_design_object;
  adora_bag : BAG OF add_design_object_request_assignment;
  pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(add_design_object_assignment) BY 1;
  REPEAT j := 1 TO SIZEOF(add_design_object_assignment[i].items) BY 1;

```

```

IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
add_design_object_assignment[i].items[j])) AND (
add_design_object_assignment[i].items[j].name =
'design object addition') THEN
IF EXISTS(add_design_object_assignment[i].items[j].
related_product_definition) THEN
IF NOT (add_design_object_assignment[i].items[j].
related_product_definition IN pd_bag) THEN
pd_bag := pd_bag + add_design_object_assignment[i].items[j].
related_product_definition;
END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(add_design_object_request_assignment) BY 1;
REPEAT j := 1 TO SIZEOF(add_design_object_request_assignment[i].
items) BY 1;
IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
add_design_object_request_assignment[i].items[j])) AND (
add_design_object_request_assignment[i].items[j].name =
'design object addition') THEN
IF EXISTS(add_design_object_request_assignment[i].items[j].
related_product_definition) THEN
IF NOT (add_design_object_request_assignment[i].items[j].
related_product_definition IN pd_bag) THEN
pd_bag := pd_bag + add_design_object_request_assignment[i].
items[j].related_product_definition;
END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pd_bag) BY 1;
IF NOT pass THEN
ESCAPE;
END_IF;
mdo_bag := [];
pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') | (
pdr.name = 'design object addition') );
REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
IF NOT pass THEN
ESCAPE;
END_IF;
adoa_bag := QUERY ( adoa <* add_design_object_assignment | (
pdr_bag[j] IN adoa.items) );
REPEAT k := 1 TO SIZEOF(adoa_bag) BY 1;
IF NOT pass THEN

```

```

        ESCAPE;
    END_IF;
    REPEAT l := 1 TO SIZEOF(adoa_bag[k].items) BY 1;
        IF EXISTS(adoa_bag[k].items[l]) THEN
            IF adoa_bag[k].items[l] IN mdo_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                mdo_bag := mdo_bag + adoa_bag[k].items[l];
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    adora_bag := QUERY ( adora <* add_design_object_request_assignment
        | (pdr_bag[j] IN adora.items) );
    REPEAT k := 1 TO SIZEOF(adora_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        REPEAT l := 1 TO SIZEOF(adora_bag[k].items) BY 1;
            IF adora_bag[k].items[l] IN mdo_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                mdo_bag := mdo_bag + adora_bag[k].items[l];
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- add_design_object_management_relationship_unique_constraint

RULE adjacent_stratum_surface_definition_constraint FOR (
    shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'adjacent stratum surface definition') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_SURFACE') IN TYPEOF(sar.related_shape_aspect)) AND (
        sar.related_shape_aspect.description IN ['secondary surface']))))))
        = 0);

```

```

WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'adjacent stratum surface definition') AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_SURFACE') IN TYPEOF(sar.relater_shape_aspect))) AND (
    sar.relater_shape_aspect.description IN ['primary surface'])))))) ))
    = 0);

```

```

END_RULE; -- adjacent_stratum_surface_definition_constraint

```

```

RULE adjacent_stratum_surface_definition_unique_constraint FOR (
    shape_aspect_relationship);

```

```

LOCAL
    sss_bag : BAG OF stratum_surface;
    pss_bag : BAG OF stratum_surface := [];
    name_bag : BAG OF STRING := [];
    sar_bag : BAG OF shape_aspect_relationship;
    pass1 : BOOLEAN := TRUE;
    pass2 : BOOLEAN := TRUE;
    assd : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.description =
        'adjacent stratum surface definition') );
END_LOCAL;
REPEAT i := 1 TO SIZEOF(assd) BY 1;
    IF EXISTS(assd[i].name) THEN
        IF assd[i].name IN name_bag THEN
            pass1 := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + assd[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(assd) BY 1;
    IF EXISTS(assd[i].relater_shape_aspect) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'STRATUM_SURFACE') IN TYPEOF(assd[i].relater_shape_aspect)
            THEN
            IF NOT (assd[i].relater_shape_aspect IN pss_bag) THEN
                pss_bag := pss_bag + assd[i].relater_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pss_bag) BY 1;
    IF NOT pass2 THEN
        ESCAPE;
    END_IF;
    sss_bag := [];
    sar_bag := QUERY ( sar <* assd | (sar.relater_shape_aspect :=:
        pss_bag[i]) );

```

```

REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
  IF EXISTS(sar_bag[j].related_shape_aspect) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM_SURFACE') IN TYPEOF(assd[j].related_shape_aspect)
    THEN
      IF sar_bag[j].related_shape_aspect IN sss_bag THEN
        pass2 := FALSE;
        ESCAPE;
      ELSE
        sss_bag := sss_bag + sar_bag[j].related_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass1;
  WR2: pass2;

END_RULE; -- adjacent_stratum_surface_definition_unique_constraint

RULE aggregate_connectivity_requirement_unique_constraint FOR (
  product_definition_relationship);

LOCAL
  acr    : BAG OF product_definition_relationship := QUERY ( pdr <*
    product_definition_relationship | (pdr.name =
      'aggregate connectivity requirement') );
  pd_bag : BAG OF product_definition := [];
  pass   : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(acr) BY 1;
  IF EXISTS(acr[i].related_product_definition) THEN
    IF acr[i].id = 'design composition path' THEN
      IF acr[i].related_product_definition IN pd_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        pd_bag := pd_bag + acr[i].related_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- aggregate_connectivity_requirement_unique_constraint

```

```
RULE analytical_model_port_unique_constraint FOR (analytical_model_port);
```

```
LOCAL
```

```
  rr_bag : BAG OF representation_relationship;
```

```
  name_bag : BAG OF STRING := [];
```

```
  amp_bag : BAG OF analytical_model_port;
```

```
  am_bag : BAG OF analytical_model;
```

```
  pass : BOOLEAN := TRUE;
```

```
END_LOCAL;
```

```
REPEAT i := 1 TO SIZEOF(analytical_model_port) BY 1;
```

```
  IF EXISTS(analytical_model_port[i].name) THEN
```

```
    IF NOT (analytical_model_port[i].name IN name_bag) THEN
```

```
      name_bag := name_bag + analytical_model_port[i].name;
```

```
    END_IF;
```

```
  END_IF;
```

```
END_REPEAT;
```

```
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
```

```
  IF NOT pass THEN
```

```
    ESCAPE;
```

```
  END_IF;
```

```
  amp_bag := QUERY ( amp <* analytical_model_port | (amp.name =  
    name_bag[i]) );
```

```
  am_bag := [];
```

```
  REPEAT j := 1 TO SIZEOF(amp_bag) BY 1;
```

```
    IF NOT pass THEN
```

```
      ESCAPE;
```

```
    END_IF;
```

```
    rr_bag := QUERY ( rr <* USEDIN(amp_bag[j],  
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +  
      'REPRESENTATION_RELATIONSHIP.REP_2') | ((rr.name =  
      'access mechanism') AND ((  
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +  
      'ANALYTICAL_MODEL') IN TYPEOF(rr.rep_1))) );
```

```
    REPEAT k := 1 TO SIZEOF(rr_bag) BY 1;
```

```
      IF EXISTS(rr_bag[k].rep_1) THEN
```

```
        IF rr_bag[k].rep_1 IN am_bag THEN
```

```
          pass := FALSE;
```

```
          ESCAPE;
```

```
        ELSE
```

```
          am_bag := am_bag + rr_bag[k].rep_1;
```

```
        END_IF;
```

```
      END_IF;
```

```
    END_REPEAT;
```

```
  END_REPEAT;
```

```
END_REPEAT;
```

```
WHERE
```

```
  WR1: pass;
```

```
END_RULE; -- analytical_model_port_unique_constraint
```

```

RULE angular_dimension_with_direction_vector_unique_constraint FOR (
    angular_dimension_with_orientation);

LOCAL
    p_bag    : BAG OF property_definition;
    pdr_bag   : BAG OF property_definition_relationship;
    pd_bag    : BAG OF property_definition := [];
    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(angular_dimension_with_orientation) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    p_bag := QUERY ( pd <* USEDIN(angular_dimension_with_orientation[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
        'dimensional location property') );
    REPEAT j := 1 TO SIZEOF(p_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(p_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
            | ((pdr.name = 'measurement orientation') AND (pdr.
            related_property_definition.description =
            'datum based vector orientation')) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].related_property_definition) THEN
                IF pdr_bag[k].related_property_definition IN pd_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    pd_bag := pd_bag + pdr_bag[k].related_property_definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- angular_dimension_with_direction_vector_unique_constraint

RULE application_context_requires_ap_definition FOR (application_context,
    application_protocol_definition);

WHERE
    WR1: (SIZEOF(QUERY ( ac <* application_context | (NOT (SIZEOF(
        QUERY ( apd <* application_protocol_definition | ((ac :=: apd.

```

```

        application) AND (apd.
        application_interpreted_model_schema_name =
        'electronic_assembly_interconnect_and_packaging_design')) )) =
        1)) )) = 0);

END_RULE; -- application_context_requires_ap_definition

RULE approval_requires_approval_date_time FOR (approval,
        approval_date_time);

WHERE
    WR1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( adt <*
        approval_date_time | (app :=: adt.dated_approval) )) = 1)) ))
        = 0));

END_RULE; -- approval_requires_approval_date_time

RULE approval_requires_approval_person_organization FOR (approval,
        approval_person_organization);

WHERE
    WR1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( apo <*
        approval_person_organization | (app :=: apo.
        authorized_approval) )) >= 1)) )) = 0);

END_RULE; -- approval_requires_approval_person_organization

RULE approvals_are_assigned FOR (approval, approval_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( aa <*
        approval_assignment | (app :=: aa.assigned_approval) )) >= 1)) ))
        = 0);

END_RULE; -- approvals_are_assigned

RULE assembly_composition_relationship_unique_constraint FOR (
        assembly_component_usage);

LOCAL
    acu_bag : BAG OF assembly_component_usage;
    acr      : BAG OF assembly_component_usage := QUERY ( acu <*
        assembly_component_usage | (acu.name =
        'assembly composition') );
    cd_bag   : BAG OF component_definition;
    pu_bag   : BAG OF physical_unit := [];
    pass     : BOOLEAN := TRUE;

```



```

END_LOCAL;
REPEAT i := 1 TO SIZEOF(acr) BY 1;
  IF EXISTS(acr[i].relating_product_definition) THEN
    IF (acr[i].relating_product_definition.frame_of_reference.name =
      'physical design') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT') IN TYPEOF(acr[i].relating_product_definition))
    THEN
      IF NOT (acr[i].relating_product_definition IN pu_bag) THEN
        pu_bag := pu_bag + acr[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  cd_bag := [];
  acu_bag := QUERY ( acu <* acr | (acu.relating_product_definition :=
    pu_bag[i]) );
  REPEAT j := 1 TO SIZEOF(acu_bag) BY 1;
    IF EXISTS(acu_bag[j].related_product_definition) THEN
      IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(acu_bag[j].
        related_product_definition) THEN
        IF acu_bag[j].related_product_definition IN cd_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          cd_bag := cd_bag + acu_bag[j].related_product_definition;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- assembly_composition_relationship_unique_constraint

RULE assembly_module_usage_view_connector_relationship_unique_constraint FOR (
  assembly_module_usage_view_connector_relationship);

LOCAL
  rr_bag   : BAG OF representation_relationship;
  pdr_bag  : BAG OF property_definition_representation;
  desc_bag : BAG OF STRING;
  pd_bag   : BAG OF property_definition;
  amuvcr   : BAG OF assembly_module_usage_view_connector_relationship;

```

```

    pu_bag    : BAG OF physical_unit := [];
    pass      : BOOLEAN := TRUE;
    r_bag     : BAG OF representation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(
    assembly_module_usage_view_connector_relationship) BY 1;
    r_bag := USEDIN(assembly_module_usage_view_connector_relationship[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS');
    REPEAT j := 1 TO SIZEOF(r_bag) BY 1;
        pdr_bag := QUERY ( pdr <* USEDIN(r_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT') IN TYPEOF(pdr.definition.definition)) AND (pdr
            .definition.definition.frame_of_reference.name =
            'physical design usage') AND (SIZEOF(QUERY ( prpc <* USEDIN(pdr
            .definition.definition.formation.of_product,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc.name =
            'assembly module') )) > 0)) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].definition.definition) THEN
                IF NOT (pdr_bag[k].definition.definition IN pu_bag) THEN
                    pu_bag := pu_bag + pdr_bag[k].definition.definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    desc_bag := [];
    pd_bag := USEDIN(pu_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION');
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := USEDIN(pd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF NOT pass THEN
                ESCAPE;
            END_IF;
            amuvcr := QUERY ( r <*
                assembly_module_usage_view_connector_relationship | (r IN
                pdr_bag[k].used_representation.items) );

```

```

REPEAT l := 1 TO SIZEOF(amuvcr) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  r_bag := USEDIN(amuvcr[l],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS');
  REPEAT m := 1 TO SIZEOF(r_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    rr_bag := QUERY ( rr <* USEDIN(r_bag[m],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
      + 'REPRESENTATION_RELATIONSHIP.REP_2') | ((rr.name =
      'reference designation assignment') AND (rr.rep_1.name =
      'externally visible partial reference designation') AND (
      ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
      + 'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(rr.rep_1.
      items[1])) AND (rr.rep_1.items[1].name =
      'partial reference designation')) );
    REPEAT n := 1 TO SIZEOF(rr_bag) BY 1;
      IF EXISTS(rr_bag[n].rep_1.items[1].description) THEN
        IF rr_bag[n].rep_1.items[1].description IN desc_bag
          THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            desc_bag := desc_bag + rr_bag[n].rep_1.items[1].
              description;
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- assembly_module_usage_view_connector_relationship_unique_constraint

RULE bare_die_unique_constraint FOR (bare_die,
  externally_defined_bare_die, library_defined_bare_die);

LOCAL
  fu_bag : BAG OF functional_unit := [];
  pdr_bag : BAG OF product_definition_relationship;
  pd_bag : BAG OF property_definition;
  ut_bag : BAG OF property_definition;

```

```

pu_bag : BAG OF physical_unit;
lddb   : BAG OF physical_unit := QUERY ( r <*
    library_defined_bare_die | (r.frame_of_reference.name =
    'physical design usage') );
pu      : BAG OF physical_unit;
bd      : BAG OF physical_unit := QUERY ( r <* bare_die | (r.
    frame_of_reference.name = 'physical design usage') );
pass    : BOOLEAN := TRUE;
edbd    : BAG OF physical_unit := QUERY ( r <*
    externally_defined_bare_die | (r.frame_of_reference.name =
    'physical design usage') );
END_LOCAL;
pu := bd + edbd + lddb;
REPEAT i := 1 TO SIZEOF(pu) BY 1;
    pdr_bag := QUERY ( pdr <* USEDIN(pu[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') | (
        (pdr.name = 'implemented function') AND (pdr.
        relating_product_definition.frame_of_reference.name =
        'functional design usage') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(pdr.relatng_product_definition))) );
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF EXISTS(pdr_bag[j].relating_product_definition) THEN
            IF NOT (pdr_bag[j].relating_product_definition IN fu_bag) THEN
                fu_bag := fu_bag + pdr_bag[j].relating_product_definition;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(fu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pu_bag := QUERY ( r <* pu | (SIZEOF(QUERY ( pdr <* USEDIN(fu_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        ((pdr.name = 'implemented function') AND (pdr.
        related_product_definition :=: r)) )) > 0) );
    pd_bag := [];
    REPEAT j := 1 TO SIZEOF(pu_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        ut_bag := QUERY ( pd <* USEDIN(pu_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') | (pd.name =
            'unit technology') );
        REPEAT k := 1 TO SIZEOF(ut_bag) BY 1;
            IF EXISTS(ut_bag[k]) THEN
                IF ut_bag[k] IN pd_bag THEN
                    pass := FALSE;
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

        ESCAPE;
    ELSE
        pd_bag := pd_bag + ut_bag[k];
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- bare_die_unique_constraint


RULE boundary_size_characteristic_constraint FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | (((rep.name =
        'diametrical boundary size') OR (rep.name =
        'opposing boundary set size')) AND (NOT (SIZEOF(
        QUERY ( lmwu <* rep.items | (lmwu.name = 'tolerance value') ))
        = 1))) )) = 0));

END_RULE; -- boundary_size_characteristic_constraint


RULE bus_element_relationship_constraint FOR (
    product_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name = 'bus element composition') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(pdr.
        relating_product_definition)))))) )) = 0);

END_RULE; -- bus_element_relationship_constraint


RULE bus_element_relationship_unique_constraint FOR (
    product_definition_relationship);

LOCAL
    pdr_bag : BAG OF product_definition_relationship;
    bsd_bag : BAG OF bus_structural_definition := [];
    pd_bag : BAG OF product_definition;
    ber : BAG OF product_definition_relationship := QUERY ( pdr <*
        product_definition_relationship | (pdr.name =
        'bus element composition') );
    pass : BOOLEAN := TRUE;

```

```

END_LOCAL;
REPEAT i := 1 TO SIZEOF(ber) BY 1;
  IF EXISTS(ber[i].relating_product_definition) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(ber[i].
        relating_product_definition) THEN
      IF NOT (ber[i].relating_product_definition IN bsd_bag) THEN
        bsd_bag := bsd_bag + ber[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(bsd_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  pdr_bag := QUERY ( pdr <* ber | (pdr.relating_product_definition :=:
    bsd_bag[i]) );
  REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(pdr_bag[i].
        related_product_definition)) OR (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'NETWORK_NODE_DEFINITION') IN TYPEOF(pdr_bag[i].
        related_product_definition)) AND (pdr_bag[i].
        related_product_definition.frame_of_reference.name =
        'functional network design')) THEN
      IF EXISTS(pdr_bag[j].related_product_definition) THEN
        IF pdr_bag[j].related_product_definition IN pd_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pd_bag := pd_bag + pdr_bag[j].related_product_definition;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- bus_element_relationship_unique_constraint

RULE bus_structural_definition_unique_constraint FOR (
  bus_structural_definition, product_definition_formation,
  product);

LOCAL
  name_bag : BAG OF STRING := [];
  pass      : BOOLEAN := TRUE;

```

```

END_LOCAL;
REPEAT i := 1 TO SIZEOF(bus_structural_definition) BY 1;
  IF EXISTS(bus_structural_definition[i].formation.of_product.name)
    THEN
      IF bus_structural_definition[i].formation.of_product.name IN
        name_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          name_bag := name_bag + bus_structural_definition[i]\
            product_definition.formation.of_product.name;
        END_IF;
      END_IF;
    END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- bus_structural_definition_unique_constraint

RULE cartesian_coordinate_system_constraint FOR (
  global_unit_assigned_context, geometric_representation_context,
  global_uncertainty_assigned_context);

WHERE
  WR1: (SIZEOF(QUERY ( guac <* global_unit_assigned_context | (NOT (
    SIZEOF(guac.units) <= 3)) )) = 0);
  WR2: (SIZEOF(QUERY ( guac <* global_unit_assigned_context | (NOT ((
    SIZEOF(QUERY ( u <* guac.units | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LENGTH_UNIT'
      IN TYPEOF(u)) )) = 1) AND (SIZEOF(QUERY ( u <* guac.units | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PLANE_ANGLE_UNIT'
      IN TYPEOF(u)) AND ((u.name = radian) OR ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONVERSION_BASED_UNIT'
      IN TYPEOF(u)) AND (u\conversion_based_unit.name = 'degree')
      AND (u\conversion_based_unit.conversion_factor.value_component
      = 0.0174533) AND (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PLANE_ANGLE_MEASURE_WITH_UNIT'
      IN TYPEOF(u\conversion_based_unit.conversion_factor)))))) )) =
      1) AND (SIZEOF(QUERY ( u <* guac.units | (
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SOLID_ANGLE_UNIT'
      IN TYPEOF(u)) )) <= 1))) )) = 0);
  WR3: (SIZEOF(QUERY ( grc <* geometric_representation_context | (NOT ((
    grc.coordinate_space_dimension = 2) OR (grc.
    coordinate_space_dimension = 3))) )) = 0);
  WR4: (SIZEOF(QUERY ( guac <* global_uncertainty_assigned_context | (

```

```

        NOT (SIZEOF(QUERY ( u <* guac.uncertainty | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(u)) )) = 1)) )) = 0);

END_RULE; -- cartesian_coordinate_system_constraint


RULE cartesian_transformation_operator_2d_constraint FOR (
    cartesian_transformation_operator_2d);

WHERE
    WR1: (SIZEOF(QUERY ( cto2 <* cartesian_transformation_operator_2d | (
        NOT (EXISTS(cto2.axis1) AND EXISTS(cto2.axis2) AND (cto2.scale
        > 0))) )) = 0);

END_RULE; -- cartesian_transformation_operator_2d_constraint


RULE certification_requires_date_or_date_and_time FOR (certification,
    applied_date_and_time_assignment, applied_date_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( cert <* certification | (NOT ((SIZEOF(
        QUERY ( adata <* applied_date_and_time_assignment | (cert IN
        adata.items) )) = 1) OR (SIZEOF(QUERY ( ada <*
        applied_date_assignment | (cert IN ada.items) )) = 1))) )) = 0);

END_RULE; -- certification_requires_date_or_date_and_time


RULE certification_unique_constraint FOR (certification);

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF certification := QUERY ( r <* certification | TRUE );
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].name) THEN
        IF r[i]\certification.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + r[i].name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- certification_unique_constraint

```



```

RULE change_design_object_management_relationship_unique_constraint FOR (
    change_from_design_object_assignment,
    change_from_design_object_request_assignment,
    change_to_design_object_assignment,
    change_to_design_object_request_assignment);

LOCAL
    pdr_bag      : BAG OF product_definition_relationship;
    pd_bag       : BAG OF product_definition := [];
    pass1        : BOOLEAN := TRUE;
    pass2        : BOOLEAN := TRUE;
    ctdora_bag   : BAG OF change_to_design_object_request_assignment;
    cfdora_bag   : BAG OF change_from_design_object_request_assignment;
    mdo_bag      : BAG OF managed_design_object;
    ctdoa_bag    : BAG OF change_to_design_object_assignment;
    cfdoa_bag    : BAG OF change_from_design_object_assignment;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(change_from_design_object_assignment) BY 1;
    REPEAT j := 1 TO SIZEOF(change_from_design_object_assignment[i].
        items) BY 1;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
                change_from_design_object_assignment[i].items[j])) AND (
                change_from_design_object_assignment[i].items[j].name =
                'design object change') THEN
            IF EXISTS(change_from_design_object_assignment[i].items[j].
                relating_product_definition) THEN
                IF NOT (change_from_design_object_assignment[i].items[j].
                    relating_product_definition IN pd_bag) THEN
                    pd_bag := pd_bag + change_from_design_object_assignment[i].
                        items[j].relating_product_definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(change_from_design_object_request_assignment)
    BY 1;
    REPEAT j := 1 TO SIZEOF(change_from_design_object_request_assignment
        [i].items) BY 1;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
                change_from_design_object_request_assignment[i].items[j])) AND
            (change_from_design_object_request_assignment[i].items[j].name
            = 'design object change') THEN
            IF EXISTS(change_from_design_object_request_assignment[i].items[
                j].relating_product_definition) THEN
                IF NOT (change_from_design_object_request_assignment[i].items[
                    j].relating_product_definition IN pd_bag) THEN
                    pd_bag := pd_bag +
                        change_from_design_object_request_assignment[i].items[j].

```

```

        relating_product_definition;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pd_bag) BY 1;
    IF NOT (pass1 AND pass2) THEN
        ESCAPE;
    END_IF;
    IF pass1 THEN
        mdo_bag := [];
        pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[i],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | (pdr.name = 'design object change') );
        REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
            IF NOT pass1 THEN
                ESCAPE;
            END_IF;
            ctdoa_bag := QUERY ( ctdoa <* change_to_design_object_assignment
                | (pdr_bag[j] IN ctdoa.items) );
            REPEAT k := 1 TO SIZEOF(ctdoa_bag) BY 1;
                IF NOT pass1 THEN
                    ESCAPE;
                END_IF;
                REPEAT l := 1 TO SIZEOF(ctdoa_bag[k].items) BY 1;
                    IF EXISTS(ctdoa_bag[k].items[l]) THEN
                        IF ctdoa_bag[k].items[l] IN mdo_bag THEN
                            pass1 := FALSE;
                            ESCAPE;
                        ELSE
                            mdo_bag := mdo_bag + ctdoa_bag[k].items[l];
                        END_IF;
                    END_IF;
                END_REPEAT;
            END_REPEAT;
        END_REPEAT;
        REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
            IF NOT pass1 THEN
                ESCAPE;
            END_IF;
            ctdora_bag := QUERY ( ctdora <*
                change_to_design_object_request_assignment | (pdr_bag[j] IN
                ctdora.items) );
            REPEAT k := 1 TO SIZEOF(ctdora_bag) BY 1;
                IF NOT pass1 THEN
                    ESCAPE;
                END_IF;
                REPEAT l := 1 TO SIZEOF(ctdora_bag[k].items) BY 1;
                    IF EXISTS(ctdora_bag[k].items[l]) THEN
                        IF ctdora_bag[k].items[l] IN mdo_bag THEN

```

```

        pass1 := FALSE;
        ESCAPE;
    ELSE
        mdo_bag := mdo_bag + ctdora_bag[k].items[l];
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_IF;
IF pass2 THEN
    mdo_bag := [];
    pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION')
        | (pdr.name = 'design object change') );
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF NOT pass2 THEN
            ESCAPE;
        END_IF;
        cfdoa_bag := QUERY ( cfdoa <*
            change_from_design_object_assignment | (pdr_bag[j] IN cfdoa.
            items) );
        REPEAT k := 1 TO SIZEOF(cfdoa_bag) BY 1;
            IF NOT pass2 THEN
                ESCAPE;
            END_IF;
            REPEAT l := 1 TO SIZEOF(cfdoa_bag[k].items) BY 1;
                IF EXISTS(cfdoa_bag[k].items[l]) THEN
                    IF cfdoa_bag[k].items[l] IN mdo_bag THEN
                        pass2 := FALSE;
                        ESCAPE;
                    ELSE
                        mdo_bag := mdo_bag + cfdoa_bag[k].items[l];
                    END_IF;
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF NOT pass2 THEN
            ESCAPE;
        END_IF;
        cfdora_bag := QUERY ( cfdora <*
            change_from_design_object_request_assignment | (pdr_bag[j] IN
            cfdora.items) );
        REPEAT k := 1 TO SIZEOF(cfdora_bag) BY 1;
            IF NOT pass2 THEN
                ESCAPE;
            END_IF;
            REPEAT l := 1 TO SIZEOF(cfdora_bag[k].items) BY 1;
                IF EXISTS(cfdora_bag[k].items[l]) THEN

```

```

        IF cfdora_bag[k].items[l] IN mdo_bag THEN
            pass2 := FALSE;
            ESCAPE;
        ELSE
            mdo_bag := mdo_bag + cfdora_bag[k].items[l];
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_IF;
END_REPEAT;

WHERE
    WR1: pass1;
    WR2: pass2;

END_RULE; -- change_design_object_management_relationship_unique_constraint

RULE change_request_unique_constraint FOR (versioned_action_request);

LOCAL
    var_bag      : BAG OF versioned_action_request;
    version_bag   : BAG OF STRING := [];
    pass          : BOOLEAN := TRUE;
    id_bag        : BAG OF STRING;
    cr            : BAG OF versioned_action_request := QUERY ( v < *
        versioned_action_request | (SIZEOF(QUERY ( ara < *
            USEDIN(v, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
                + 'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST')
                | (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
                + 'CHANGE_REQUEST') IN TYPEOF(ara)) )) > 0) ));
END_LOCAL;
REPEAT i := 1 TO SIZEOF(versioned_action_request) BY 1;
    IF EXISTS(versioned_action_request[i].version) THEN
        IF NOT (versioned_action_request[i].version IN version_bag) THEN
            version_bag := version_bag + versioned_action_request[i].version;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(version_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    var_bag := QUERY ( v < * versioned_action_request | (v.version =
        version_bag[i]) );
    id_bag := [];
    REPEAT j := 1 TO SIZEOF(var_bag) BY 1;
        IF EXISTS(var_bag[j].id) THEN
            IF var_bag[j].id IN id_bag THEN
                pass := FALSE;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

        ESCAPE;
    ELSE
        id_bag := id_bag + var_bag[j].id;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- change_request_unique_constraint

RULE characteristic_unique_constraint FOR (representation_item);

LOCAL
    name_bag : BAG OF STRING := [];
    ri_bag   : BAG OF representation_item;
    adr_bag  : BAG OF applied_document_reference;
    pass     : BOOLEAN := TRUE;
    d_bag    : BAG OF document;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(representation_item) BY 1;
    IF EXISTS(representation_item[i].name) THEN
        IF NOT (representation_item[i].name IN name_bag) THEN
            name_bag := name_bag + representation_item[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    ri_bag := QUERY ( ri <* representation_item | (ri.name = name_bag[i]) );
    d_bag := [];
    REPEAT j := 1 TO SIZEOF(ri_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        adr_bag := USEDIN(ri_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS');
        REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
            IF EXISTS(adr_bag[k].assigned_document) THEN
                IF adr_bag[k].assigned_document IN d_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    d_bag := d_bag + adr_bag[k].assigned_document;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- characteristic_unique_constraint

RULE compatible_dimension FOR (cartesian_point, direction,
    representation_context, geometric_representation_context);

WHERE
    WR1: (SIZEOF(QUERY ( x <* cartesian_point | (SIZEOF(QUERY ( y <*
        geometric_representation_context | (item_in_context(x,y) AND (
        HIINDEX(x.coordinates) <> y.coordinate_space_dimension)) )) >
        0) )) = 0);
    WR2: (SIZEOF(QUERY ( x <* direction | (SIZEOF(QUERY ( y <*
        geometric_representation_context | (item_in_context(x,y) AND (
        HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension)) ))
        > 0) )) = 0);

END_RULE; -- compatible_dimension

RULE component_external_reference_constraint FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'component external reference') AND (SIZEOF(QUERY ( dri <* rep
        .items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'design owner')))) )) = 1)) )) = 0);
    WR2: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'component external reference') AND (SIZEOF(QUERY ( dri <* rep
        .items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'part number')))) )) = 1)) )) = 0);
    WR3: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'component external reference') AND (SIZEOF(QUERY ( dri <* rep
        .items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'revision code')))) )) = 1)) )) = 0);
    WR4: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'component external reference') AND (SIZEOF(QUERY ( dri <* rep
        .items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.

```

```

        name = 'product definition id')))) )) = 1)) )) = 0);
WR5: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'component external reference') AND (SIZEOF(QUERY ( dri <* rep
        .items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'reference designation')))) )) = 1)) )) = 0);

END_RULE; -- component_external_reference_constraint

```

```

RULE component_feature_external_reference_unique_constraint FOR (
    descriptive_representation_item);

LOCAL
    desc_bag : BAG OF STRING := [];
    dri_bag : BAG OF descriptive_representation_item;
    cfer : BAG OF descriptive_representation_item := QUERY ( dri <*
        descriptive_representation_item | (dri.name =
        'component feature external reference') );
    rep_bag : BAG OF representation;
    pass : BOOLEAN := TRUE;
    r_bag : BAG OF representation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(cfer) BY 1;
    IF EXISTS(cfer[i].description) THEN
        IF NOT (cfer[i].description IN desc_bag) THEN
            desc_bag := desc_bag + cfer[i].description;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(desc_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    r_bag := [];
    dri_bag := QUERY ( dri <* cfer | (dri.description = desc_bag[i]) );
    REPEAT j := 1 TO SIZEOF(dri_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        rep_bag := QUERY ( r <* USEDIN(dri_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION.ITEMS') | (r.name =
            'component external reference') );
        REPEAT k := 1 TO SIZEOF(rep_bag) BY 1;
            IF EXISTS(rep_bag[k]) THEN
                IF rep_bag[k] IN r_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    r_bag := r_bag + rep_bag[k];
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

        END_IF;
        END_IF;
        END_REPEAT;
        END_REPEAT;
        END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- component_feature_external_reference_unique_constraint

RULE component_feature_to_layout_feature_relationship_constraint FOR (
    shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'component feature to layout feature') AND (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL'] * TYPEOF(sar.relatng_shape_aspect)) > 0))) ))
        = 0);
    WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'component feature to layout feature') AND (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE'] * TYPEOF(sar.related_shape_aspect)) > 0))) ))
        = 0);

END_RULE; -- component_feature_to_layout_feature_relationship_constraint

RULE component_feature_unique_constraint FOR (component_terminal,
    component_interface_terminal);

LOCAL
    cd_bag : BAG OF component_definition := [];
    sar_bag : BAG OF shape_aspect_relationship;
    cit      : BAG OF shape_aspect := QUERY ( cit <*
        component_interface_terminal | (cit.description IN [
        'interconnect component interface terminal',
        'packaged connector component interface terminal']) );
    def_bag : BAG OF shape_aspect;
    pass    : BOOLEAN := TRUE;
    cf       : BAG OF shape_aspect;
    ct       : BAG OF shape_aspect := QUERY ( ct <* component_terminal |
        (ct.description IN ['assembly module component terminal',
        'bare die component terminal',

```



```

        'interconnect component join terminal',
        'interconnect module component terminal',
        'interconnect module component stratum based terminal',
        'minimally defined component terminal',
        'packaged component join terminal']) );
    sa_bag : BAG OF shape_aspect;
END_LOCAL;
cf := ct + cit;
REPEAT i := 1 TO SIZEOF(cf) BY 1;
    IF EXISTS(cf[i].of_shape.definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPONENT_DEFINITION') IN TYPEOF(cf[i].of_shape.definition)
            THEN
                IF NOT (cf[i].of_shape.definition IN cd_bag) THEN
                    cd_bag := cd_bag + cf[i].of_shape.definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
REPEAT i := 1 TO SIZEOF(cd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    def_bag := [];
    sa_bag := QUERY ( sa <* cf | (sa.of_shape.definition :=: cd_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        sar_bag := QUERY ( sar <* USEDIN(sa_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.name =
            'instantiated feature') );
        REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
            IF EXISTS(sar_bag[k].relating_shape_aspect) THEN
                IF sar_bag[k].relating_shape_aspect IN def_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    def_bag := def_bag + sar_bag[k].relating_shape_aspect;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- component_feature_unique_constraint

```

```
RULE component_mounting_clearance_relationship_constraint FOR (
    representation);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'mounting clearance') AND (SIZEOF(QUERY ( lmwu <* rep.items |
        ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'minimum height')) )) = 0)) )) = 0);
```

```
END_RULE; -- component_mounting_clearance_relationship_constraint
```

```
RULE component_overlap_relationship_constraint FOR (representation);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'overlap clearance') AND (SIZEOF(QUERY ( lmwu <* rep.items | (
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'minimum clearance')) )) = 0)) )) = 0);
```

```
END_RULE; -- component_overlap_relationship_constraint
```

```
RULE component_part_2d_geometric_representation_relationship_constraint FOR (
    shape_representation_relationship);
```

```
    LOCAL
```

```
        rr : SET OF representation_relationship := QUERY ( srr <*
            shape_representation_relationship | (srr\
            representation_relationship.name =
            'component part planar shape') );
```

```
    END_LOCAL;
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( srr_t <* rr | (NOT (srr_t.rep_1 :=: srr_t.rep_2)) ))
        = 0);
```

```
END_RULE; -- component_part_2d_geometric_representation_relationship_constraint
```

```
RULE component_part_3d_geometric_representation_relationship_constraint FOR (
    shape_representation_relationship);
```

```
    LOCAL
```

```
        rr : SET OF representation_relationship := QUERY ( srr <*
            shape_representation_relationship | (srr\
            representation_relationship.name = 'component part 3d shape') );
```

```
    END_LOCAL;
```

```

WHERE
  WR1: (SIZEOF(QUERY ( srr_t <* rr | (NOT (srr_t.rep_1 :=: srr_t.rep_2)) ))
        = 0);

END_RULE;

-- component_part_3d_geometric_representation_relationship_constraint

RULE component_placement_restriction_assignment_constraint FOR (
  property_definition_representation);

WHERE
  WR1: (SIZEOF(QUERY ( pdr <* property_definition_representation | ((pdr
    .used_representation.name =
    'component placement restriction assignment') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_AREA') IN TYPEOF(pdr.definition.
    definition)))))) )) = 0);
  WR2: (SIZEOF(QUERY ( pdr <* property_definition_representation | ((pdr
    .used_representation.name =
    'component placement restriction assignment') AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUPED_REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.definition))
    AND (pdr.definition\group.name =
    'item restricted requirements property')))) )) = 0);

END_RULE; -- component_placement_restriction_assignment_constraint

RULE component_terminal_to_assembly_module_terminal_assignment_unique_constraint FOR (
  shape_aspect_relationship);

LOCAL
  cttamta : BAG OF shape_aspect_relationship := QUERY ( sar <*
    shape_aspect_relationship | (sar.name = 'design usage') );
  amt_bag : BAG OF assembly_module_terminal := [];
  pass    : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(cttamta) BY 1;
  IF EXISTS(cttamta[i].relating_shape_aspect) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(cttamta[i].
      relating_shape_aspect) THEN
      IF cttamta[i].relating_shape_aspect IN amt_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        amt_bag := amt_bag + cttamta[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

```

```

WHERE
    WR1: pass;

END_RULE;

-- component_terminal_to_assembly_module_terminal_assignment_unique_constraint

RULE
component_terminal_to_interconnect_module_terminal_assignment_unique_constraint
FOR ( shape_aspect_relationship);

LOCAL
    imt_bag : BAG OF interconnect_module_terminal := [];
    pass     : BOOLEAN := TRUE;
    cttimta  : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.name = 'design usage') );
END_LOCAL;
REPEAT i := 1 TO SIZEOF(cttimta) BY 1;
    IF EXISTS(cttimta[i].relating_shape_aspect) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(cttimta[i].
                relating_shape_aspect) THEN
            IF cttimta[i].relating_shape_aspect IN imt_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                imt_bag := imt_bag + cttimta[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE;

-- component_terminal_to_interconnect_module_terminal_assignment_unique_constraint

RULE component_terminal_to_pca_terminal_assignment_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    cttamta : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.name = 'design usage') );
    amt_bag : BAG OF assembly_module_terminal := [];
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(cttamta) BY 1;
    IF EXISTS(cttamta[i].relating_shape_aspect) THEN
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(cttamta[i].
        relating_shape_aspect)) AND (cttamta[i].relating_shape_aspect.
        description = 'pca terminal') THEN
    IF cttamta[i].relating_shape_aspect IN amt_bag THEN
        pass := FALSE;
        ESCAPE;
    ELSE
        amt_bag := amt_bag + cttamta[i].relating_shape_aspect;
    END_IF;
END_IF;
END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- component_terminal_to_pca_terminal_assignment_unique_constraint

RULE component_termination_passage_template_terminal_unique_constraint FOR (
    shape_aspect);

LOCAL
    name_bag : BAG OF STRING := [];
    sar_bag : BAG OF shape_aspect_relationship;
    pass : BOOLEAN := TRUE;
    ptd_bag : BAG OF part_template_definition;
    ctptt : BAG OF shape_aspect := QUERY ( sa <* shape_aspect | ((sa.
        description =
        'component termination passage template interface terminal')
        OR (sa.description =
        'component termination passage template join terminal')) );
    sa_bag : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ctptt) BY 1;
    IF EXISTS(ctptt[i].name) THEN
        IF NOT (ctptt[i].name IN name_bag) THEN
            name_bag := name_bag + ctptt[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    ptd_bag := [];
    sa_bag := QUERY ( sa <* ctptt | (sa.name = name_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        sar_bag := QUERY ( sar <* USEDIN(sa_bag[j],

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
= 'access mechanism') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PART_TEMPLATE_DEFINITION') IN TYPEOF(sar.relying_shape_aspect))
AND (sar.relying_shape_aspect.name =
        'component termination passage template')) );
REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[k].relating_shape_aspect) THEN
        IF sar_bag[k].relating_shape_aspect IN ptd_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            ptd_bag := ptd_bag + sar_bag[k].relating_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- component_termination_passage_template_terminal_unique_constraint


RULE composite_shape_element_constraint FOR (composite_shape_aspect);

WHERE
    WR1: (SIZEOF(QUERY ( csa <* composite_shape_aspect | (SIZEOF(
        QUERY ( sar <* USEDIN(csa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.name
        = 'composing') )) < 2) )) = 0);

END_RULE; -- composite_shape_element_constraint


RULE conductive_interconnect_element_terminal_unique_constraint FOR (
    component_terminal);

LOCAL
    name_bag : BAG OF STRING := [];
    csa_bag : BAG OF component_shape_aspect;
    sar_bag : BAG OF shape_aspect_relationship;
    ciet : BAG OF component_terminal := QUERY ( ct <*
        component_terminal | (ct.description =
        'conductive interconnect element terminal') );
    ct_bag : BAG OF component_terminal;
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ciet) BY 1;

```

```

    IF EXISTS(ciet[i].name) THEN
      IF NOT (ciet[i].name IN name_bag) THEN
        name_bag := name_bag + ciet[i].name;
      END_IF;
    END_IF;
  END_REPEAT;
  REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    ct_bag := QUERY ( ct <* ciet | (ct.name = name_bag[i]) );
    csa_bag := [];
    REPEAT j := 1 TO SIZEOF(ct_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      sar_bag := QUERY ( sar <* USEDIN(ct_bag[j],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
        = 'associated component') AND (sar.relater_shape_aspect.
        description IN [
          'conductive interconnect element with pre defined transitions',
          'conductive interconnect element with user defined single transition'])
        AND (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'COMPONENT_SHAPE_ASPECT') IN TYPEOF(sar.relater_shape_aspect))) );
      REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[k].relater_shape_aspect) THEN
          IF sar_bag[k].relater_shape_aspect IN csa_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            csa_bag := csa_bag + sar_bag[k].relater_shape_aspect;
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- conductive_interconnect_element_terminal_unique_constraint

RULE configuration_item_requires_person_organization FOR (
  configuration_item, applied_person_and_organization_assignment,
  applied_organization_assignment);

WHERE
  WR1: (SIZEOF(QUERY ( ci <* configuration_item | (NOT ((SIZEOF(
    QUERY ( apaoa <* applied_person_and_organization_assignment |
    (ci IN apaoa.items) )) = 1) OR (SIZEOF(QUERY ( aoa <*

```

```

        applied_organization_assignment | (ci IN aoa.items) )) = 1))) ))
        = 0);

END_RULE; -- configuration_item_requires_person_organization

RULE connection_zone_constraint FOR (shape_aspect, representation);

WHERE
  WR1: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
    'connection zone') AND (NOT (SIZEOF(QUERY ( pd <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (sdr.
    used_representation.context_of_items.context_type =
    'connection zone colour') )) = 1) )) = 1))) )) = 0);
  WR2: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
    'connection zone') AND (NOT (SIZEOF(QUERY ( pd <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( sdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_REPRESENTATION') IN TYPEOF(sdr.used_representation))
    AND (sdr.used_representation.name = 'zone shape')) )) = 1) ))
    = 1))) )) = 0);
  WR3: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
    'connection zone') AND (NOT (SIZEOF(QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.name
    IN ['terminal connection zone', 'external connection zone',
    'external connection area', 'minimum attachment region size']) ))
    <= 1))) )) = 0);
  WR4: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
    'connection zone') AND (NOT (SIZEOF(QUERY ( sar <* USEDIN(sa,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.name
    IN ['internal connection zone']) )) <= 1))) )) = 0);

END_RULE; -- connection_zone_constraint

RULE connectivity_allocation_constraint FOR (
  product_definition_relationship);

WHERE
  WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
    name = 'connectivity allocation') AND (SIZEOF(QUERY ( pds <*
    USEDIN(pdr,

```



```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION') | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(pds)) AND (SIZEOF(
QUERY ( pcd <* USEDIN(pds,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(pcd)) )) >= 1)) ))
= 0)) )) = 0);

```

END_RULE; -- connectivity_allocation_constraint

RULE connectivity_allocation_unique_constraint FOR (
product_definition_relationship);

```

LOCAL
pd_bag : BAG OF property_definition;
pass : BOOLEAN := TRUE;
ca : BAG OF product_definition_relationship := QUERY ( pdr <*
product_definition_relationship | (pdr.name =
'connectivity allocation') );
pcd_bag : BAG OF physical_connectivity_definition := [];
sa_bag : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ca) BY 1;
IF NOT pass THEN
ESCAPE;
END_IF;
pd_bag := USEDIN(ca[i].related_product_definition,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION.DEFINITION');
REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
IF NOT pass THEN
ESCAPE;
END_IF;
sa_bag := QUERY ( sa <* USEDIN(pd_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT.OF_SHAPE') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(sa)) );
REPEAT k := 1 TO SIZEOF(sa_bag) BY 1;
IF EXISTS(sa_bag[k]) THEN
IF sa_bag[k] IN pcd_bag THEN
pass := FALSE;
ESCAPE;
ELSE
pcd_bag := pcd_bag + sa_bag[k];
END_IF;
END_IF;
END_REPEAT;

```

```

        END_REPEAT;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- connectivity_allocation_unique_constraint


RULE contract_constraint FOR (contract);

WHERE
    WR1: (SIZEOF(QUERY ( c <* contract | (SIZEOF(USEDIN(c,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 0) )) = 0);

END_RULE; -- contract_constraint


RULE contract_unique_constraint FOR (contract);

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF contract := QUERY ( r <* contract | TRUE );
    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].name) THEN
        IF r[i]\contract.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + r[i].name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- contract_unique_constraint


RULE coordinated_assembly_and_shape FOR (next_assembly_usage_occurrence);

WHERE
    WR1: (SIZEOF(QUERY ( nauo <* next_assembly_usage_occurrence | (NOT
        assembly_shape_is_defined(nauo,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN')
        ) )) =

```

```

        0);

END_RULE; -- coordinated_assembly_and_shape

RULE curve_style_font_and_scaling_unique_constraint FOR (
    curve_style_font_and_scaling);

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF curve_style_font_and_scaling := QUERY ( r <*
        curve_style_font_and_scaling | TRUE );
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].name) THEN
        IF r[i]\curve_style_font_and_scaling.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + r[i].name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- curve_style_font_and_scaling_unique_constraint

RULE curve_style_requires_length_measure_with_unit FOR (curve_style);

WHERE
    WR1: (SIZEOF(QUERY ( cs <* curve_style | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(cs))) )) = 0);

END_RULE; -- curve_style_requires_length_measure_with_unit

RULE curve_style_unique_constraint FOR (representation);

LOCAL
    desc_bag : BAG OF STRING := [];
    pass     : BOOLEAN := TRUE;
    cs       : BAG OF representation := QUERY ( r <* representation | ((
        r.name = 'curve style parameters') AND (r.
        context_of_items.context_type =
        'curve style parametric context')) );
END_LOCAL;

```

```

REPEAT i := 1 TO SIZEOF(cs) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  REPEAT j := 1 TO SIZEOF(cs[i].items) BY 1;
    IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(cs[i].items[j]))
      AND (cs[i].items[j].name = 'curve style name') THEN
      IF EXISTS(cs[i].items[j].description) THEN
        IF cs[i].items[j].description IN desc_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          desc_bag := desc_bag + cs[i].items[j].description;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- curve_style_unique_constraint


RULE date_and_time_require_minute_and_second FOR (date_and_time);

WHERE
  WR1: (SIZEOF(QUERY ( dat <* date_and_time | (NOT EXISTS(dat.
    time_component.minute_component)) )) = 0);
  WR2: (SIZEOF(QUERY ( dat <* date_and_time | (NOT EXISTS(dat.
    time_component.second_component)) )) = 0);

END_RULE; -- date_and_time_require_minute_and_second


RULE date_constraint FOR (date);

WHERE
  WR1: (SIZEOF(QUERY ( d <* date | (d.year_component < 1856) )) = 0);

END_RULE; -- date_constraint


RULE datum_feature_usage_in_datum_relationship_unique_constraint FOR (
  shape_aspect_relationship);

LOCAL
  pud_bag : BAG OF physical_unit_datum;
  pudf_bag : BAG OF physical_unit_datum_feature := [];

```

```

    sar_bag : BAG OF shape_aspect_relationship;
    dfuidr   : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | ((sar.name =
            'datum feature usage in common datum') OR (sar.name =
            'datum feature usage in single datum')) );
    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dfuidr) BY 1;
    IF EXISTS(dfuidr[i].relating_shape_aspect) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT_DATUM') IN TYPEOF(dfuidr[i].
                relating_shape_aspect) THEN
            IF NOT (dfuidr[i].relating_shape_aspect IN pud_bag) THEN
                pud_bag := pud_bag + dfuidr[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pud_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pudf_bag := [];
    sar_bag := QUERY ( sar <* dfuidr | ((sar.relating_shape_aspect :=:
        pud_bag[i]) AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT_DATUM_FEATURE') IN TYPEOF(sar.related_shape_aspect))) );
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF sar_bag[j].related_shape_aspect IN pudf_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                pudf_bag := pudf_bag + sar_bag[j].related_shape_aspect;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- datum_feature_usage_in_datum_relationship_unique_constraint

RULE datum_precedence_assignment_unique_constraint FOR (
    property_definition);

LOCAL
    pdr_bag : BAG OF property_definition_relationship;
    name_bag : BAG OF STRING := [];
    pd_bag   : BAG OF property_definition;

```

```

    dpa      : BAG OF property_definition := QUERY ( pd <*
        property_definition | (pd.description =
            'datum precedence assignment') );
    pass     : BOOLEAN := TRUE;
    pds      : BAG OF property_definition;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dpa) BY 1;
    IF EXISTS(dpa[i].name) THEN
        IF NOT (dpa[i].name IN name_bag) THEN
            name_bag := name_bag + dpa[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pds := [];
    pd_bag := QUERY ( pd <* dpa | (pd.name = name_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
            | ((pdr.name = 'assigned datum precedence') AND (pdr.
                relating_property_definition.description IN [
                    'datum system property with material conditions',
                    'datum system property without material conditions']))) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].relating_property_definition) THEN
                IF pdr_bag[k].relating_property_definition IN pds THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    pds := pds + pdr_bag[k].relating_property_definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- datum_precedence_assignment_unique_constraint

RULE datum_target_usage_in_datum_target_set_relationship_unique_constraint
    FOR ( shape_aspect_relationship);

```

```

LOCAL
  sar_bag    : BAG OF shape_aspect_relationship;
  pass1      : BOOLEAN := TRUE;
  pass2      : BOOLEAN := TRUE;
  dtuidts    : BAG OF shape_aspect_relationship := QUERY ( sar <*
    shape_aspect_relationship | (sar.name =
    'datum target usage') );
  pudt_bag   : BAG OF physical_unit_datum_target;
  pudts_bag  : BAG OF physical_unit_datum_target_set := [];
  num_bag    : BAG OF STRING;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dtuidts) BY 1;
  IF EXISTS(dtuidts[i].relating_shape_aspect) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_DATUM_TARGET') IN TYPEOF(dtuidts[i].
    relating_shape_aspect) THEN
      IF NOT (dtuidts[i].relating_shape_aspect IN pudts_bag) THEN
        pudts_bag := pudts_bag + dtuidts[i].relating_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pudts_bag) BY 1;
  IF NOT (pass1 AND pass2) THEN
    ESCAPE;
  END_IF;
  num_bag := [];
  pudt_bag := [];
  sar_bag := QUERY ( sar <* dtuidts | (sar.relating_shape_aspect :=:
    pudts_bag[i]) );
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF NOT (pass1 AND pass2) THEN
      ESCAPE;
    END_IF;
    IF pass1 AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_UNIT_DATUM_TARGET') IN TYPEOF(sar_bag[j].
      related_shape_aspect)) THEN
      IF EXISTS(sar_bag[j].related_shape_aspect) THEN
        IF sar_bag[j].related_shape_aspect IN pudt_bag THEN
          pass1 := FALSE;
        ELSE
          pudt_bag := pudt_bag + sar_bag[j].related_shape_aspect;
        END_IF;
      END_IF;
    END_IF;
    IF pass2 THEN
      IF EXISTS(sar_bag[j].description) THEN
        IF sar_bag[j].description IN num_bag THEN
          pass2 := FALSE;
        ELSE
          num_bag := num_bag + sar_bag[j].description;
        END_IF;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

```

```

        END_IF;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass1;
    WR2: pass2;

END_RULE;

-- datum_target_usage_in_datum_target_set_relationship_unique_constraint

RULE datum_usage_in_datum_system_relationship_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    ds_bag : BAG OF datum_system := [];
    pud_bag : BAG OF physical_unit_datum;
    sar_bag : BAG OF shape_aspect_relationship;
    duidsr : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.name =
            'datum usage in datum system') );
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(duidsr) BY 1;
    IF EXISTS(duidsr[i].relating_shape_aspect) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DATUM_SYSTEM') IN TYPEOF(duidsr[i].relating_shape_aspect)
            THEN
            IF NOT (duidsr[i].relating_shape_aspect IN ds_bag) THEN
                ds_bag := ds_bag + duidsr[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(ds_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pud_bag := [];
    sar_bag := QUERY ( sar <* duidsr | (sar.relating_shape_aspect :=:
        ds_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'PHYSICAL_UNIT_DATUM') IN TYPEOF(sar_bag[j].
                related_shape_aspect) THEN
                IF sar_bag[j].related_shape_aspect IN pud_bag THEN
                    pass := FALSE;
                    ESCAPE;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```



```

        ELSE
            pud_bag := pud_bag + sar_bag[j].related_shape_aspect;
        END_IF;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- datum_usage_in_datum_system_relationship_unique_constraint

RULE default_assembly_bond_shape_constraint FOR (shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'default assembly bond shape') AND (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(sr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') IN
        TYPEOF(pdr.definition.definition)) )) = 1))) )) = 0);

END_RULE; -- default_assembly_bond_shape_constraint

RULE default_passage_based_land_physical_template_unique_constraint FOR (
    land_physical_template);

LOCAL
    pdr_bag : BAG OF property_definition_relationship;
    sar_bag : BAG OF shape_aspect_relationship;
    st_bag : BAG OF stratum_technology;
    dpblpt : BAG OF land_physical_template := QUERY ( lpt <*
        land_physical_template | (lpt\shape_aspect.description IN
        ['default via based',
        'default attachment size and via based',
        'default unsupported passage based',
        'default component termination passage based',
        'default attachment size and component termination passage based']) );
    pt_bag : BAG OF passage_technology := [];
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dpblpt) BY 1;
    sar_bag := QUERY ( sar <* USEDIN(dpblpt[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.name =
        'technology usage') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PASSAGE_TECHNOLOGY') IN TYPEOF(sar.related_shape_aspect))) );
REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
  IF EXISTS(sar_bag[j].related_shape_aspect) THEN
    IF NOT (sar_bag[j].related_shape_aspect IN pt_bag) THEN
      pt_bag := pt_bag + sar_bag[j].related_shape_aspect;
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pt_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  st_bag := [];
  sar_bag := QUERY ( sar <* USEDIN(pt_bag[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name =
    'technology usage') AND (sar.relateing_shape_aspect IN dpblpt)) );
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pdr_bag := QUERY ( pdr <* USEDIN(sar_bag[j].relating_shape_aspect.
      of_shape,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
      | ((pdr.name = 'technology usage') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM_TECHNOLOGY') IN TYPEOF(pdr.
      relating_property_definition.definition)))) );
    REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
      IF EXISTS(pdr_bag[k].relating_property_definition.definition)
        THEN
          IF pdr_bag[k].relating_property_definition.definition IN
            st_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            st_bag := st_bag + pdr_bag[k].relating_property_definition.
              definition;
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- default_passage_based_land_physical_template_unique_constraint

```

```

RULE delete_design_object_management_relationship_unique_constraint FOR (
    delete_design_object_assignment,
    delete_design_object_request_assignment);

LOCAL
    pdr_bag    : BAG OF product_definition_relationship := [];
    ddoa_bag    : BAG OF delete_design_object_assignment;
    pd_bag      : BAG OF product_definition := [];
    ddora_bag   : BAG OF delete_design_object_request_assignment;
    mdo_bag     : BAG OF managed_design_object;
    pass        : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(delete_design_object_assignment) BY 1;
    REPEAT j := 1 TO SIZEOF(delete_design_object_assignment[i].items)
        BY 1;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
                delete_design_object_assignment[i].items[j])) AND (
                delete_design_object_assignment[i].items[j].name =
                'design object deletion') THEN
            IF EXISTS(delete_design_object_assignment[i].items[j].
                relating_product_definition) THEN
                IF NOT (delete_design_object_assignment[i].items[j].
                    relating_product_definition IN pd_bag) THEN
                    pd_bag := pd_bag + delete_design_object_assignment[i].items[
                        j].relating_product_definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(delete_design_object_request_assignment) BY 1;
    REPEAT j := 1 TO SIZEOF(delete_design_object_request_assignment[i].
        items) BY 1;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(
                delete_design_object_request_assignment[i].items[j])) AND (
                delete_design_object_request_assignment[i].items[j].name =
                'design object deletion') THEN
            IF EXISTS(delete_design_object_request_assignment[i].items[j].
                relating_product_definition) THEN
                IF NOT (delete_design_object_request_assignment[i].items[j].
                    relating_product_definition IN pd_bag) THEN
                    pd_bag := pd_bag + delete_design_object_request_assignment[i].
                        items[j].relating_product_definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pd_bag) BY 1;
    IF NOT pass THEN

```

```

    ESCAPE;
  END_IF;
  mdo_bag := [];
  pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    (pdr.name = 'design object deletion') );
  REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    ddoa_bag := QUERY ( ddoa <* delete_design_object_assignment | (
      pdr_bag[j] IN ddoa.items) );
    REPEAT k := 1 TO SIZEOF(ddoa_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      REPEAT l := 1 TO SIZEOF(ddoa_bag[k].items) BY 1;
        IF EXISTS(ddoa_bag[k].items[l]) THEN
          IF ddoa_bag[k].items[l] IN mdo_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            mdo_bag := mdo_bag + ddoa_bag[k].items[l];
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;
  REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    ddora_bag := QUERY ( ddora <*
      delete_design_object_request_assignment | (pdr_bag[j] IN ddora.
      items) );
    REPEAT k := 1 TO SIZEOF(ddora_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      REPEAT l := 1 TO SIZEOF(ddora_bag[k].items) BY 1;
        IF EXISTS(ddora_bag[k].items[l]) THEN
          IF ddora_bag[k].items[l] IN mdo_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            mdo_bag := mdo_bag + ddora_bag[k].items[l];
          END_IF;
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

```

```

END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- delete_design_object_management_relationship_unique_constraint

RULE dependent_instantiable_action_directive FOR (action_directive);

WHERE
  WR1: (SIZEOF(QUERY ( ad <* action_directive | (NOT (SIZEOF(USEDIN(ad,
    '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_action_directive

RULE dependent_instantiable_address FOR (address);

WHERE
  WR1: (SIZEOF(QUERY ( add <* address | (NOT (SIZEOF(USEDIN(add, '')) >=
    1)) )) = 0);

END_RULE; -- dependent_instantiable_address

RULE dependent_instantiable_approval_status FOR (approval_status);

WHERE
  WR1: (SIZEOF(QUERY ( ast <* approval_status | (NOT (SIZEOF(USEDIN(ast,
    '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_approval_status

RULE dependent_instantiable_certification_type FOR (certification_type);

WHERE
  WR1: (SIZEOF(QUERY ( ct <* certification_type | (NOT (SIZEOF(USEDIN(ct,
    '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_certification_type

RULE dependent_instantiable_contract_type FOR (contract_type);

WHERE
  WR1: (SIZEOF(QUERY ( ct <* contract_type | (NOT (SIZEOF(USEDIN(ct, ''))
    >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_contract_type

```

```

RULE dependent_instantiable_date FOR (date);

WHERE
  WR1: (SIZEOF(QUERY ( dt <* date | (NOT (SIZEOF(USEDIN(dt, '')) >= 1)) ))
        = 0);

END_RULE; -- dependent_instantiable_date

```

```

RULE dependent_instantiable_date_role FOR (date_role);

WHERE
  WR1: (SIZEOF(QUERY ( dr <* date_role | (NOT (SIZEOF(USEDIN(dr, '')) >=
        1)) )) = 0);

END_RULE; -- dependent_instantiable_date_role

```

```

RULE dependent_instantiable_date_time_role FOR (date_time_role);

WHERE
  WR1: (SIZEOF(QUERY ( dtr <* date_time_role | (NOT (SIZEOF(USEDIN(dtr,
        '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_date_time_role

```

```

RULE dependent_instantiable_document_type FOR (document_type);

WHERE
  WR1: (SIZEOF(QUERY ( dt <* document_type | (NOT (SIZEOF(USEDIN(dt, ''))
        >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_document_type

```

```

RULE dependent_instantiable_named_unit FOR (named_unit);

WHERE
  WR1: (SIZEOF(QUERY ( nu <* named_unit | (NOT (SIZEOF(USEDIN(nu, '')) >=
        1)) )) = 0);

END_RULE; -- dependent_instantiable_named_unit

```

```

RULE dependent_instantiable_organization_role FOR (organization_role);

WHERE
  WR1: (SIZEOF(QUERY ( org <* organization_role | (NOT (SIZEOF(USEDIN(

```

```

        org, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_organization_role

RULE dependent_instantiable_parametric_representation_context FOR (
    parametric_representation_context);

WHERE
    WR1: (SIZEOF(QUERY ( prc <* parametric_representation_context | (NOT (
        SIZEOF(USEDIN(prc, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_parametric_representation_context

RULE dependent_instantiable_person_and_organization_role FOR (
    person_and_organization_role);

WHERE
    WR1: (SIZEOF(QUERY ( poar <* person_and_organization_role | (NOT (
        SIZEOF(USEDIN(poar, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_person_and_organization_role

RULE dependent_instantiable_representation_item FOR (representation_item);

WHERE
    WR1: (SIZEOF(QUERY ( ri <* representation_item | (NOT (SIZEOF(USEDIN(
        ri, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_representation_item

RULE dependent_instantiable_security_classification_level FOR (
    security_classification_level);

WHERE
    WR1: (SIZEOF(QUERY ( scl <* security_classification_level | (NOT (
        SIZEOF(USEDIN(scl, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_security_classification_level

RULE dependent_instantiable_shape_representation FOR (
    shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT (SIZEOF(USEDIN(
        sr, '')) >= 1)) )) = 0);

```

```

END_RULE; -- dependent_instantiable_shape_representation

RULE dependent_instantiable_tolerance_value FOR (tolerance_value);

WHERE
  WR1: (SIZEOF(QUERY ( tv <* tolerance_value | (NOT (SIZEOF(USEDIN(tv, ''))
    >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_tolerance_value

RULE dependently_located_layer_connection_point_unique_constraint FOR (
  layer_connection_point);

LOCAL
  sar_bag : BAG OF shape_aspect_relationship;
  s_bag   : BAG OF stratum := [];
  lcp_bag : BAG OF layer_connection_point;
  pass    : BOOLEAN := TRUE;
  dllcp   : BAG OF layer_connection_point := QUERY ( lcp <*
    layer_connection_point | (lcp.description =
    'dependently located') );
  sa_bag  : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dllcp) BY 1;
  IF EXISTS(dllcp[i].of_shape.definition) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'STRATUM') IN TYPEOF(dllcp[i].of_shape.definition) THEN
      IF NOT (dllcp[i].of_shape.definition IN s_bag) THEN
        s_bag := s_bag + dllcp[i].of_shape.definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(s_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  lcp_bag := QUERY ( lcp <* dllcp | (lcp.of_shape.definition :=: s_bag
    [i]) );
  sa_bag := [];
  REPEAT j := 1 TO SIZEOF(lcp_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    sar_bag := QUERY ( sar <* USEDIN(lcp_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
      = 'associated design object') AND (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

        'INTER_STRATUM_FEATURE') IN TYPEOF(sar.relater_shape_aspect))
    OR (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_TERMINAL') IN TYPEOF(sar.
        relater_shape_aspect)))) );
REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[k].relater_shape_aspect) THEN
        IF sar_bag[k].relater_shape_aspect IN sa_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            sa_bag := sa_bag + sar_bag[k].relater_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- dependently_located_layer_connection_point_unique_constraint


RULE design_functional_unit_allocation_to_assembly_component_constraint
    FOR ( product_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name =
            'design functional unit allocation to assembly component') AND
            (NOT ((pdr.relater_product_definition.description =
            'design definition path') AND (pdr.relater_product_definition
            .id = 'design composition path')))) )) = 0);
    WR2: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name =
            'design functional unit allocation to assembly component') AND
            (NOT ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPONENT_DEFINITION') IN TYPEOF(pdr.
            relater_product_definition)))) )) = 0);

END_RULE;

-- design_functional_unit_allocation_to_assembly_component_constraint


RULE design_functional_unit_allocation_to_assembly_component_unique_constraint
    FOR ( product_definition_relationship);

LOCAL
    pdr_bag : BAG OF product_definition_relationship;
    dfuatac : BAG OF product_definition_relationship := QUERY ( pdr <*
        product_definition_relationship | (pdr.name =

```

```

        'design functional unit allocation to assembly component') );
    cd_bag : BAG OF component_definition := [];
    pass   : BOOLEAN := TRUE;
    id_bag : BAG OF STRING;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dfuatat) BY 1;
    IF EXISTS(dfuatat[i].related_product_definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPONENT_DEFINITION') IN TYPEOF(dfuatat[i].
            related_product_definition) THEN
            IF NOT (dfuatat[i].related_product_definition IN cd_bag) THEN
                cd_bag := cd_bag + dfuatat[i].related_product_definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(cd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pdr_bag := QUERY ( pdr <* dfuatat | (pdr.related_product_definition
        :=: cd_bag[i]) );
    id_bag := [];
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF EXISTS(pdr_bag[j].id) THEN
            IF pdr_bag[j].id IN id_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                id_bag := id_bag + pdr_bag[j].id;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE;

-- design_functional_unit_allocation_to_assembly_component_unique_constraint

RULE design_functional_unit_allocation_to_assembly_joint_unique_constraint
    FOR ( product_definition_relationship);

LOCAL
    dfuataj : BAG OF product_definition_relationship := QUERY ( pdr <*
        product_definition_relationship | (pdr.name =
        'design functional unit allocation to assembly joint') );
    pd_bag : BAG OF product_definition := [];
    pass   : BOOLEAN := TRUE;
END_LOCAL;

```

```

REPEAT i := 1 TO SIZEOF(dfuataj) BY 1;
  IF EXISTS(dfuataj[i].relating_product_definition) THEN
    IF dfuataj[i].relating_product_definition.id =
      'design composition path' THEN
      IF dfuataj[i].relating_product_definition IN pd_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        pd_bag := pd_bag + dfuataj[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE;

-- design_functional_unit_allocation_to_assembly_joint_unique_constraint

RULE design_functional_unit_allocation_to_fabrication_joint_unique_constraint
  FOR ( product_definition_relationship);

LOCAL
  dfuatfj : BAG OF product_definition_relationship := QUERY ( pdr < *
    product_definition_relationship | (pdr.name =
      'design functional unit allocation to fabrication joint') );
  pd_bag : BAG OF product_definition := [];
  pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dfuatfj) BY 1;
  IF EXISTS(dfuatfj[i].relating_product_definition) THEN
    IF dfuatfj[i].relating_product_definition.id =
      'design composition path' THEN
      IF dfuatfj[i].relating_product_definition IN pd_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        pd_bag := pd_bag + dfuatfj[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE;

-- design_functional_unit_allocation_to_fabrication_joint_unique_constraint

```

```
RULE design_intent_modification_in_stratum_3d_position_constraint FOR (
    mapped_item);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
    'positioned design intent modification shape in stratum') AND
    (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AXIS2_PLACEMENT_3D') IN TYPEOF(mi.mapping_target)))))) = 0);
```

```
END_RULE; -- design_intent_modification_in_stratum_3d_position_constraint
```

```
RULE design_intent_modification_in_stratum_planar_position_constraint FOR (
    mapped_item);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
    'design intent modification in stratum planar position') AND (
    NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'AXIS2_PLACEMENT_2D') IN TYPEOF(mi.mapping_target)))))) = 0);
WR2: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
    'design intent modification in stratum planar position') AND (
    SIZEOF(QUERY ( rep <* USEDIN(mi,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS') | (SIZEOF(USEDIN(rep,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')) = 0) ))
    = 0)) )) = 0);
WR3: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
    'design intent modification in stratum planar position') AND (
    NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'AXIS2_PLACEMENT_2D') IN TYPEOF(mi.mapping_source.
    mapped_representation)))))) = 0);
```

```
END_RULE; -- design_intent_modification_in_stratum_planar_position_constraint
```

```
RULE design_intent_modification_sequence_constraint FOR (
    shape_aspect_relationship);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'shape modification sequence') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_MODIFICATION') IN TYPEOF(sar.relate_shape_aspect)))))) = 0);
WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'shape modification sequence') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_MODIFICATION') IN TYPEOF(sar.related_shape_aspect)))))) = 0);
```

```
END_RULE; -- design_intent_modification_sequence_constraint
```

```
RULE design_intent_modification_sequence_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    dims      : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.name =
            'shape modification sequence') );
    sar_bag    : BAG OF shape_aspect_relationship;
    pm_bag     : BAG OF shape_modification := [];
    pass       : BOOLEAN := TRUE;
    sm_bag     : BAG OF shape_modification;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dims) BY 1;
    IF EXISTS(dims[i].relating_shape_aspect) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_MODIFICATION') IN TYPEOF(dims[i].relating_shape_aspect)
            THEN
                IF NOT (dims[i].relating_shape_aspect IN pm_bag) THEN
                    pm_bag := pm_bag + dims[i].relating_shape_aspect;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
REPEAT i := 1 TO SIZEOF(pm_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sar_bag := QUERY ( sar <* dims | (sar.relating_shape_aspect :=:
        pm_bag[i]) );
    sm_bag := [];
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'SHAPE_MODIFICATION') IN TYPEOF(sar_bag[j].
                    related_shape_aspect) THEN
                IF sar_bag[j].related_shape_aspect IN sm_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    sm_bag := sm_bag + sar_bag[j].related_shape_aspect;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;
```

```
END_RULE; -- design_intent_modification_sequence_unique_constraint
```

```
RULE device_terminal_map_relationship_constraint FOR (
    shape_aspect_relationship);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'device terminal map relationship') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DEVICE_TERMINAL_MAP') IN TYPEOF(sar.relatng_shape_aspect)))) )
        = 0);
```

```
    WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'device terminal map relationship') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DEVICE_TERMINAL_MAP') IN TYPEOF(sar.related_shape_aspect)))) )
        = 0);
```

```
END_RULE; -- device_terminal_map_relationship_constraint
```

```
RULE dimension_along_curve_unique_constraint FOR (
    dimensional_location_with_path);
```

```
LOCAL
```

```
    pass : BOOLEAN := TRUE;
```

```
    sa_bag : BAG OF shape_aspect := [];
```

```
END_LOCAL;
```

```
REPEAT i := 1 TO SIZEOF(dimensional_location_with_path) BY 1;
```

```
    IF EXISTS(dimensional_location_with_path[i].path) THEN
```

```
        IF dimensional_location_with_path[i].path IN sa_bag THEN
```

```
            pass := FALSE;
```

```
            ESCAPE;
```

```
        ELSE
```

```
            sa_bag := sa_bag + dimensional_location_with_path[i].path;
```

```
        END_IF;
```

```
    END_IF;
```

```
END_REPEAT;
```

```
WHERE
```

```
    WR1: pass;
```

```
END_RULE; -- dimension_along_curve_unique_constraint
```

```
RULE dimension_location_with_direction_vector_unique_constraint FOR (
    dimensional_location_with_direction);
```

```
LOCAL
```

```
    p_bag : BAG OF property_definition;
```

```

pdr_bag    : BAG OF property_definition_relationship;
pdrep_bag  : BAG OF property_definition_representation;
pd_bag     : BAG OF property_definition := [];
place_bag  : BAG OF placement := [];
pass       : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dimensional_location_with_direction) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  p_bag := QUERY ( pd <* USEDIN(dimensional_location_with_direction[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
    'dimensional location property') );
  REPEAT j := 1 TO SIZEOF(p_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pdr_bag := QUERY ( pdr <* USEDIN(p_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
      | ((pdr.name = 'measurement orientation') AND (pdr.
        related_property_definition.description =
        'datum based vector orientation')) );
    REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
      IF EXISTS(pdr_bag[k].related_property_definition) THEN
        IF pdr_bag[k].related_property_definition IN pd_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pd_bag := pd_bag + pdr_bag[k].related_property_definition;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
  REPEAT j := 1 TO SIZEOF(p_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pdrep_bag := USEDIN(p_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
    REPEAT k := 1 TO SIZEOF(pdrep_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      REPEAT l := 1 TO SIZEOF(pdrep_bag[k].used_representation.items)
        BY 1;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
          + 'PLACEMENT') IN TYPEOF(pdrep_bag[k].used_representation.
            items[l])) AND (pdrep_bag[k].used_representation.items[l].
              name = 'orientation') THEN

```

```

        IF EXISTS(pdrep_bag[k].used_representation.items[l]) THEN
            IF pdrep_bag[k].used_representation.items[l] IN place_bag
            THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                place_bag := place_bag + pdrep_bag[k].
                    used_representation.items[l];
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- dimension_location_with_direction_vector_unique_constraint


RULE directed_action_requires_approval FOR (directed_action,
    applied_approval_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( da <* directed_action | (NOT (SIZEOF(
        QUERY ( aaa <* applied_approval_assignment | (da IN aaa.items) ))
        = 1)) )) = 0));

END_RULE; -- directed_action_requires_approval


RULE directed_action_requires_date_or_date_and_time FOR (directed_action,
    applied_date_and_time_assignment, applied_date_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( da <* directed_action | (NOT ((SIZEOF(
        QUERY ( adata <* applied_date_and_time_assignment | ((da IN
        adata.items) AND (adata.role.name = 'start date')) )) = 1) OR
        (SIZEOF(QUERY ( ada <* applied_date_assignment | ((da IN ada.
        items) AND (ada.role.name = 'start date')) )) = 1))) )) = 0));

END_RULE; -- directed_action_requires_date_or_date_and_time


RULE directed_axis_constraint FOR (derived_shape_aspect);

WHERE
    WR1: (SIZEOF(QUERY ( dsa <* derived_shape_aspect | ((dsa.description =
        'directed axis') AND (NOT (SIZEOF(QUERY ( sar <* USEDIN(dsa,

```



```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'derived axis') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CENTRE_OF_SYMMETRY') IN TYPEOF(sar.name.related_shape_aspect))
AND (sar.name.related_shape_aspect.description = 'axis')) )) =
        1))) )) = 0);
WR2: (SIZEOF(QUERY ( dsa <* derived_shape_aspect | ((dsa.description =
        'directed axis') AND (NOT (SIZEOF(QUERY ( sar <* USEDIN(dsa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'end shape aspect') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT') IN TYPEOF(sar.name.related_shape_aspect)))) ))
        = 1))) )) = 0);
WR3: (SIZEOF(QUERY ( dsa <* derived_shape_aspect | ((dsa.description =
        'directed axis') AND (NOT (SIZEOF(QUERY ( sar <* USEDIN(dsa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.
name = 'start shape aspect') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT') IN TYPEOF(sar.name.related_shape_aspect)))) ))
        = 1))) )) = 0);

END_RULE; -- directed_axis_constraint

```

RULE directed_axis_unique_constraint FOR (derived_shape_aspect);

```

LOCAL
    cos_bag : BAG OF centre_of_symmetry := [];
    sar_bag : BAG OF shape_aspect_relationship;
    da      : BAG OF derived_shape_aspect := QUERY ( r <*
        derived_shape_aspect | (r.description = 'directed axis') );
    pass    : BOOLEAN := TRUE;
    sar2_bag : BAG OF shape_aspect_relationship;
    sa_bag  : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(da) BY 1;
    sar_bag := QUERY ( sar <* USEDIN(da[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.name =
        'derived axis') AND (sar.related_shape_aspect.description =
        'axis') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CENTRE_OF_SYMMETRY') IN TYPEOF(sar.related_shape_aspect)))) );
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF NOT (sar_bag[j].related_shape_aspect IN cos_bag) THEN
                cos_bag := cos_bag + sar_bag[j].related_shape_aspect;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```

    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(cos_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sa_bag := [];
    sar_bag := QUERY ( sar <* USEDIN(cos_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name =
        'derived axis') AND (sar.relate_shape_aspect IN da)) );
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        sar2_bag := QUERY ( sar <* USEDIN(sar_bag[j].relate_shape_aspect,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.name
            = 'start shape aspect') );
        REPEAT k := 1 TO SIZEOF(sar2_bag) BY 1;
            IF EXISTS(sar2_bag[k].related_shape_aspect) THEN
                IF sar2_bag[k].related_shape_aspect IN sa_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    sa_bag := sa_bag + sar2_bag[k].related_shape_aspect;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- directed_axis_unique_constraint

```

RULE direction_element_constraint FOR (property_definition);

```

WHERE
    WR1: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
        'direction element') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.name = 'element direction vector') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION') IN TYPEOF(pdr.
        relate_property_definition)) AND (pdr.
        relate_property_definition.description =
        'datum direction_property')) ) = 1))) ) = 0);
    WR2: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =

```

```

        'direction element') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.name = 'half datum plane direction vector') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION') IN TYPEOF(pdr.
        related_property_definition)) AND (pdr.
        related_property_definition.description =
        'datum direction_property')) )) = 1))) )) = 0);
WR3: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
        'direction element') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.name = 'direction component') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION') IN TYPEOF(pdr.
        related_property_definition)) AND (pdr.
        related_property_definition.description =
        'datum based vector orientation')) )) = 1))) )) = 0);

END_RULE; -- direction_element_constraint

RULE edge_segment_vertex_constraint FOR (shape_aspect_relationship,
        edge_segment_vertex);

LOCAL
    sar : BAG OF shape_aspect_relationship := [];
    esv : BAG OF edge_segment_vertex := QUERY ( e <*
        edge_segment_vertex | (e <> []) );
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(esv) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sar := QUERY ( sar <* shape_aspect_relationship | (esv[i] = sar\
        shape_aspect_relationship.related_shape_aspect) );
    pass := SIZEOF(sar) = 2;
    REPEAT j := 2 TO SIZEOF(sar) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pass := sar[j] :<>: sar[j - 1];
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pass := TYPEOF(sar[j]) = TYPEOF(sar[j - 1]);
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pass := sar[j].relating_shape_aspect :=: sar[j - 1].

```

```

        relating_shape_aspect;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- edge_segment_vertex_constraint

RULE edge_segment_vertex_unique_constraint FOR (edge_segment_vertex);

LOCAL
    pdr_bag : BAG OF property_definition_representation;
    desc_bag : BAG OF STRING := [];
    pd_bag : BAG OF property_definition;
    esv : BAG OF edge_segment_vertex := QUERY ( r <*
        edge_segment_vertex | (r.name = 'single datum') );
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(esv) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pd_bag := QUERY ( pd <* USEDIN(esv[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (pd.description =
        'shape element characterization') );
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := USEDIN(pd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION');
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF NOT pass THEN
                ESCAPE;
            END_IF;
            IF (pdr_bag[k].used_representation.name =
                'shape element characterization') AND ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(pdr_bag[k].
                used_representation.items[1])) THEN
                IF EXISTS(pdr_bag[k].used_representation.items[1].description)
                THEN
                    IF pdr_bag[k].used_representation.items[1].description IN
                        desc_bag THEN
                        pass := FALSE;
                        ESCAPE;
                    ELSE
                        desc_bag := desc_bag + pdr_bag[k].used_representation.

```

```

            items[1].description;
        END_IF;
    END_IF;
    END_IF;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- edge_segment_vertex_unique_constraint

RULE ee_colour_unique_constraint FOR (colour_rgb, colour_specification);

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF colour_rgb := QUERY ( r <* colour_rgb | TRUE );
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].name) THEN
        IF r[i]\colour_specification.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + r[i].name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_colour_unique_constraint

RULE ee_document_constraint FOR (document);

WHERE
    WR1: (SIZEOF(QUERY ( doc <* document | (SIZEOF(QUERY ( adata <*
        USEDIN(doc,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (adata.role.name =
        'creation date') )) = 0) )) = 0);
    WR2: (SIZEOF(QUERY ( doc <* document | (NOT ((SIZEOF(QUERY ( aoa <*
        USEDIN(doc,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (aoa.role.name =
        'publisher') )) = 1) OR (SIZEOF(QUERY ( apaoa <* USEDIN(doc,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apaoa.
        role.name = 'publisher') )) = 1))) )) = 0);

```

```

END_RULE; -- ee_document_constraint

```

```

RULE ee_document_identifier_unique_constraint FOR (document);

```

```

LOCAL
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(document) BY 1;
    IF EXISTS(document[i].id) THEN
        IF document[i].id IN id_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            id_bag := id_bag + document[i].id;
        END_IF;
    END_IF;
END_REPEAT;

```

```

WHERE
    WR1: pass;

```

```

END_RULE; -- ee_document_identifier_unique_constraint

```

```

RULE ee_document_unique_constraint FOR (document);

```

```

LOCAL
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(document) BY 1;
    IF EXISTS(document[i].id) THEN
        IF document[i].id IN id_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            id_bag := id_bag + document[i].id;
        END_IF;
    END_IF;
END_REPEAT;

```

```

WHERE
    WR1: pass;

```

```

END_RULE; -- ee_document_unique_constraint

```

```

RULE ee_material_unique_constraint FOR (material_designation);

LOCAL
  name_bag : BAG OF STRING := [];
  adr_bag  : BAG OF applied_document_reference;
  doc_bag  : BAG OF ee_specification;
  pass     : BOOLEAN := TRUE;
  md_bag   : BAG OF material_designation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(material_designation) BY 1;
  IF EXISTS(material_designation[i].name) THEN
    IF NOT (material_designation[i].name IN name_bag) THEN
      name_bag := name_bag + material_designation[i].name;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  md_bag := QUERY ( md <* material_designation | (md.name = name_bag[i]) );
  doc_bag := [];
  REPEAT j := 1 TO SIZEOF(md_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    adr_bag := QUERY ( adr <* USEDIN(md_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DOCUMENT') IN TYPEOF(adr.assigned_document)) );
    REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
      IF EXISTS(adr_bag[k].assigned_document) THEN
        IF adr_bag[k].assigned_document IN doc_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          doc_bag := doc_bag + adr_bag[k].assigned_document;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- ee_material_unique_constraint

RULE ee_product_configuration_unique_constraint FOR (configuration_item);

```

```

LOCAL
  cd_bag : BAG OF configuration_design;
  ci_bag : BAG OF configuration_item;
  pass   : BOOLEAN := TRUE;
  id_bag : BAG OF STRING := [];
  pdf_bag : BAG OF product_definition_formation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(configuration_item) BY 1;
  IF EXISTS(configuration_item[i].id) THEN
    IF NOT (configuration_item[i].id IN id_bag) THEN
      id_bag := id_bag + configuration_item[i].id;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(id_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  pdf_bag := [];
  ci_bag := QUERY ( ci <* configuration_item | (ci.id = id_bag[i]) );
  REPEAT j := 1 TO SIZEOF(ci_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    cd_bag := USEDIN(ci_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CONFIGURATION_DESIGN.CONFIGURATION');
    REPEAT k := 1 TO SIZEOF(cd_bag) BY 1;
      IF EXISTS(cd_bag[k].design) THEN
        IF cd_bag[k].design IN pdf_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pdf_bag := pdf_bag + cd_bag[k].design;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- ee_product_configuration_unique_constraint

RULE ee_product_constraint FOR (product);

WHERE
  WR1: (SIZEOF(QUERY ( prod <* product | (NOT ((SIZEOF(QUERY ( aoa <*
    USEDIN(prod,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```



```

        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (aoa.role.name =
        'design owner') )) = 1) OR (SIZEOF(QUERY ( apaoa <* USEDIN(
        prod, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apaoa
        .role.name = 'design owner') )) = 1))) )) = 0);
WR2: (SIZEOF(QUERY ( p <* product | (SIZEOF(USEDIN(p,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT')) = 0) )) = 0);
WR3: (SIZEOF(QUERY ( p <* product | (SIZEOF(USEDIN(p,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')) < 3) )) = 0);

END_RULE; -- ee_product_constraint

```

```

RULE ee_product_definition_constraint FOR (
    product_definition_with_associated_documents);

```

WHERE

```

WR1: (SIZEOF(QUERY ( pdwad <*
    product_definition_with_associated_documents | (NOT (SIZEOF(
    QUERY ( adata <* USEDIN(pdwad,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (adata.role.name =
    'creation date') )) = 1)) )) = 0);
WR2: (SIZEOF(QUERY ( pdwad <*
    product_definition_with_associated_documents | (NOT ((SIZEOF(
    QUERY ( aoa <* USEDIN(pdwad,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (aoa.role.name =
    'creator') )) = 1) OR (SIZEOF(QUERY ( apaoa <* USEDIN(pdwad,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (apaoa.
    role.name = 'creator') )) = 1))) )) = 0);
WR3: (SIZEOF(QUERY ( pdwad <*
    product_definition_with_associated_documents | (NOT (SIZEOF(
    USEDIN(pdwad,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 1)) )) = 0);
WR4: (SIZEOF(QUERY ( pdwad <*
    product_definition_with_associated_documents | (NOT (SIZEOF(
    USEDIN(pdwad,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 1)) ))
    = 0);

```

```

END_RULE; -- ee_product_definition_constraint

```

```

RULE ee_product_definition_unique_constraint FOR (
    product_definition_with_associated_documents, physical_unit,
    functional_unit);

```

```

LOCAL
    epd      : BAG OF product_definition;
    lcs_bag  : BAG OF STRING;
    pd_bag   : BAG OF product_definition;
    fu       : BAG OF product_definition := QUERY ( r <* functional_unit
        | TRUE );
    pass1    : BOOLEAN := TRUE;
    pass2    : BOOLEAN := TRUE;
    pu       : BAG OF product_definition := QUERY ( r <* physical_unit |
        TRUE );
    pdwad    : BAG OF product_definition := QUERY ( r <*
        product_definition_with_associated_documents | TRUE );
    id_bag   : BAG OF STRING;
    pdf_bag  : BAG OF product_definition_formation;
END_LOCAL;
epd := pdwad + pu + fu;
REPEAT i := 1 TO SIZEOF(epd) BY 1;
    IF EXISTS(epd[i].formation) THEN
        IF NOT (epd[i].formation IN pdf_bag) THEN
            pdf_bag := pdf_bag + epd[i].formation;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pdf_bag) BY 1;
    IF NOT (pass1 OR pass2) THEN
        ESCAPE;
    END_IF;
    lcs_bag := [];
    id_bag := [];
    pd_bag := QUERY ( pd <* epd | (pd.formation :=: pdf_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT (pass1 OR pass2) THEN
            ESCAPE;
        END_IF;
        IF pass1 THEN
            IF EXISTS(pd_bag[j].frame_of_reference.life_cycle_stage) THEN
                IF pd_bag[j].frame_of_reference.life_cycle_stage IN lcs_bag
                    THEN
                    pass1 := FALSE;
                    ESCAPE;
                ELSE
                    lcs_bag := lcs_bag + pd_bag[j].frame_of_reference.
                        life_cycle_stage;
                END_IF;
            END_IF;
        END_IF;
        IF pass2 THEN
            IF EXISTS(pd_bag[j].id) THEN
                IF pd_bag[j].id IN id_bag THEN
                    pass2 := FALSE;
                    ESCAPE;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```

        ELSE
            id_bag := id_bag + pd_bag[j].id;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass1;
    WR2: pass2;

END_RULE; -- ee_product_definition_unique_constraint

RULE ee_product_model_unique_constraint FOR (product_concept);

LOCAL
    r      : BAG OF product_concept := QUERY ( r <* product_concept |
        TRUE );
    pass   : BOOLEAN := TRUE;
    id_bag : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].id) THEN
        IF r[i]\product_concept.id IN id_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            id_bag := id_bag + r[i].id;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_product_model_unique_constraint

RULE ee_product_unique_constraint FOR (product);

LOCAL
    aoa_bag : BAG OF applied_organization_assignment;
    p_bag   : BAG OF product;
    pao_bag : BAG OF person_and_organization;
    o_bag   : BAG OF organization;
    apaoa_bag : BAG OF applied_person_and_organization_assignment;
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(product) BY 1;

```

```

IF EXISTS(product[i].id) THEN
  IF NOT (product[i].id IN id_bag) THEN
    id_bag := id_bag + product[i].id;
  END_IF;
END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(id_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  o_bag := [];
  pao_bag := [];
  p_bag := QUERY ( p <* product | (p.id = id_bag[i]) );
  REPEAT j := 1 TO SIZEOF(p_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    aoa_bag := QUERY ( aoa <* USEDIN(p_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ORGANIZATION') IN TYPEOF(aoa.assigned_organization)) AND (aoa.
      role.name = 'design owner')) );
    REPEAT k := 1 TO SIZEOF(aoa_bag) BY 1;
      IF EXISTS(aoa_bag[k].assigned_organization) THEN
        IF aoa_bag[k].assigned_organization IN o_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          o_bag := o_bag + aoa_bag[k].assigned_organization;
        END_IF;
      END_IF;
    END_REPEAT;
    apaoa_bag := QUERY ( apaoa <* USEDIN(p_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PERSON_AND_ORGANIZATION') IN TYPEOF(apaoa.
      assigned_person_and_organization)) AND (apaoa.role.name =
      'design owner')) );
    REPEAT k := 1 TO SIZEOF(apaoa_bag) BY 1;
      IF EXISTS(apaoa_bag[k].assigned_person_and_organization) THEN
        IF apaoa_bag[k].assigned_person_and_organization IN pao_bag
          THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          pao_bag := pao_bag + apaoa_bag[k].
            assigned_person_and_organization;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

```

```

        END_REPEAT;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_product_unique_constraint


RULE ee_product_version_constraint FOR (product_definition_formation);

WHERE
    WR1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (SIZEOF(
        QUERY ( aoa <* USEDIN(pdf,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | ((aoa.role.name =
            'product_supplier') OR (aoa.role.name = 'design_supplier')) ))
        > 0) )) > 0);

END_RULE; -- ee_product_version_constraint


RULE ee_product_version_unique_constraint FOR (
    product_definition_formation);

LOCAL
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(product_definition_formation) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    REPEAT j := i + 1 TO SIZEOF(product_definition_formation) BY 1;
        IF (product_definition_formation[i].id =
            product_definition_formation[j].id) AND (
            product_definition_formation[i].of_product :=:
            product_definition_formation[j].of_product) THEN
            pass := FALSE;
            ESCAPE;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_product_version_unique_constraint


RULE ee_rule_conclusion_definition_unique_constraint FOR (
    rule_conclusion_definition, representation,
```

```

        representation_relationship);

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF representation := QUERY ( r <* representation | (r
        \representation.name = 'rule definition') );
    rr       : BAG OF representation_relationship := [];
    pass     : BOOLEAN := TRUE;
    count    : INTEGER := 0;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    count := 0;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    rr := USEDIN(r[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1');
    REPEAT j := 1 TO SIZEOF(rr) BY 1;
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'RULE_CONCLUSION_DEFINITION') IN TYPEOF(rr[j].rep_2) THEN
            IF EXISTS(rr[j].rep_2\representation.name) THEN
                IF rr[j].rep_2\representation.name IN name_bag THEN
                    pass := FALSE;
                ELSE
                    name_bag := name_bag + rr[j].rep_2\representation.name;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_rule_conclusion_definition_unique_constraint

RULE ee_rule_definition_constraint FOR (representation);

WHERE
    WR1: ((SIZEOF(QUERY ( rep <* representation | (rep.name =
        'rule definition') )) = 0) OR (SIZEOF(QUERY ( rep <*
        representation | (SIZEOF(QUERY ( dri <* rep.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'identifier')) )) = 1) )) = 1));
    WR2: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'rule definition') AND (SIZEOF(QUERY ( pdr <* USEDIN(rep,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'RULE_DEFINITION') IN TYPEOF(pdr.definition.definition)) )) =
        0)) )) = 0);

END_RULE; -- ee_rule_definition_constraint

RULE ee_rule_definition_unique_constraint FOR (representation);

LOCAL
    pdr_bag : BAG OF property_definition_representation;
    rd_bag  : BAG OF rule_definition := [];
    pd_bag  : BAG OF property_definition;
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING;
    erd     : BAG OF representation := QUERY ( r <* representation | (r.
        name = 'rule definition') );
END_LOCAL;
REPEAT i := 1 TO SIZEOF(erd) BY 1;
    pdr_bag := QUERY ( pdr <* USEDIN(erd[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_DEFINITION') IN TYPEOF(pdr.definition.definition)) );
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF EXISTS(pdr_bag[j].definition.definition) THEN
            IF NOT (pdr_bag[j].definition.definition IN rd_bag) THEN
                rd_bag := rd_bag + pdr_bag[j].definition.definition;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(rd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    id_bag := [];
    pd_bag := USEDIN(rd_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION');
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation IN erd) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF NOT pass THEN
                ESCAPE;
            END_IF;
            REPEAT l := 1 TO SIZEOF(pdr_bag[k].used_representation.items)

```

```

        BY 1;
    IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(pdr_bag[k].
            used_representation.items[l])) AND (pdr_bag[k].
            used_representation.items[l]\representation_item.name =
            'identifier') THEN
        IF EXISTS(pdr_bag[k].used_representation.items[l].
            description) THEN
            IF pdr_bag[k].used_representation.items[l].description IN
                id_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                id_bag := id_bag + pdr_bag[k].used_representation.items[
                    l].description;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_rule_definition_unique_constraint

```

```

RULE ee_rule_function_unique_constraint FOR (rule_function_definition);

```

```

LOCAL
    name_bag : BAG OF STRING := [];
    adr_bag : BAG OF applied_document_reference;
    doc_bag : BAG OF document;
    pass : BOOLEAN := TRUE;
    rfd_bag : BAG OF rule_function_definition;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(rule_function_definition) BY 1;
    IF EXISTS(rule_function_definition[i].name) THEN
        IF NOT (rule_function_definition[i].name IN name_bag) THEN
            name_bag := name_bag + rule_function_definition[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    doc_bag := [];
    rfd_bag := QUERY ( am <* rule_function_definition | (am.name =
        name_bag[i]) );

```



```

REPEAT j := 1 TO SIZEOF(rfd_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  adr_bag := QUERY ( adr <* USEDIN(rfd_bag[j],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (adr.assigned_document.
    kind.product_data_type = 'reference document') );
  REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
    IF EXISTS(adr_bag[k].assigned_document) THEN
      IF adr_bag[k].assigned_document IN doc_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        doc_bag := doc_bag + adr_bag[k].assigned_document;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- ee_rule_function_unique_constraint

RULE ee_rule_premise_definition_unique_constraint FOR (
  rule_premise_definition, representation,
  representation_relationship);

LOCAL
  name_bag : BAG OF STRING := [];
  r        : BAG OF representation := QUERY ( r <* representation | (r
    \representation.name = 'rule definition') );
  rr       : BAG OF representation_relationship := [];
  pass     : BOOLEAN := TRUE;
  count    : INTEGER := 0;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
  count := 0;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  rr := USEDIN(r[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_RELATIONSHIP.REP_1');
  REPEAT j := 1 TO SIZEOF(rr) BY 1;
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_PREMISE_DEFINITION') IN TYPEOF(rr[j].rep_2) THEN
      IF EXISTS(rr[j].rep_2\representation.name) THEN
        IF rr[j].rep_2\representation.name IN name_bag THEN

```

```

        pass := FALSE;
    ELSE
        name_bag := name_bag + rr[j].rep_2\representation.name;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_rule_premise_definition_unique_constraint

RULE ee_rule_unique_constraint FOR (rule_definition);

LOCAL
    rd_bag : BAG OF rule_definition;
    ii_bag : BAG OF STRING := [];
    si_bag : BAG OF STRING;
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(rule_definition) BY 1;
    IF EXISTS(rule_definition[i]\externally_defined_item.item_id) THEN
        IF NOT (rule_definition[i]\externally_defined_item.item_id IN
            ii_bag) THEN
            ii_bag := ii_bag + rule_definition[i]\externally_defined_item.
                item_id;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(ii_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    si_bag := [];
    rd_bag := QUERY ( rd <* rule_definition | (rd\
        externally_defined_item.item_id = ii_bag[i]) );
    REPEAT j := 1 TO SIZEOF(rd_bag) BY 1;
        IF EXISTS(rd_bag[j]\externally_defined_item.source.source_id)
            THEN
            IF rd_bag[j]\externally_defined_item.source.source_id IN si_bag
                THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                si_bag := si_bag + rd_bag[j]\externally_defined_item.source.
                    source_id;
            END_IF;
        END_IF;
    END_REPEAT;

```

```

END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- ee_rule_unique_constraint

RULE explicitly_located_layer_connection_point_unique_constraint FOR (
    layer_connection_point);

LOCAL
    pdr_bag : BAG OF property_definition_representation;
    pd_bag  : BAG OF property_definition;
    s_bag   : BAG OF stratum := [];
    cp_bag  : BAG OF cartesian_point;
    lcp_bag : BAG OF layer_connection_point;
    ellcp   : BAG OF layer_connection_point := QUERY ( lcp <*
        layer_connection_point | (lcp.description =
            'explicitly located') );
    pass    : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ellcp) BY 1;
    IF EXISTS(ellcp[i].of_shape.definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'STRATUM') IN TYPEOF(ellcp[i].of_shape.definition) THEN
            IF NOT (ellcp[i].of_shape.definition IN s_bag) THEN
                s_bag := s_bag + ellcp[i].of_shape.definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(s_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    cp_bag := [];
    lcp_bag := QUERY ( lcp <* ellcp | (lcp.of_shape.definition :=: s_bag
        [i]) );
    REPEAT j := 1 TO SIZEOF(lcp_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pd_bag := USEDIN(lcp_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION');
        REPEAT k := 1 TO SIZEOF(pd_bag) BY 1;
            IF NOT pass THEN
                ESCAPE;
            END_IF;
            pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[k],
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
        used_representation.name = 'connection point location') AND (
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CARTESIAN_POINT') IN TYPEOF(pdr.used_representation.items[1]))) );
    REPEAT l := 1 TO SIZEOF(pdr_bag) BY 1;
    IF EXISTS(pdr_bag[l].used_representation.items[1]) THEN
        IF pdr_bag[l].used_representation.items[1] IN cp_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            cp_bag := cp_bag + pdr_bag[l].used_representation.items[1];
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- explicitly_located_layer_connection_point_unique_constraint


RULE feature_group_relationship_constraint FOR (
    shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'feature group relationship') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_SHAPE_ASPECT') IN TYPEOF(sar.relying_shape_aspect)))) )
        = 0));

END_RULE; -- feature_group_relationship_constraint


RULE feature_shape_occurrence_constraint FOR (shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'feature shape') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(sr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT') IN TYPEOF(pdr.definition.definition)) )) > 1)))) )
        = 0);
    WR2: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'datum shape') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(sr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SHAPE_ASPECT') IN TYPEOF(pdr.definition.definition)) )) > 1))) ))
    = 0);
WR3: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
    'zone shape') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(sr,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT') IN TYPEOF(pdr.definition.definition)) )) > 1))) ))
    = 0);

END_RULE; -- feature_shape_occurrence_constraint

```

```

RULE fill_area_occurrence_constraint FOR (
    annotation_fill_area_occurrence);

```

```

WHERE
    WR1: (SIZEOF(QUERY ( afao <* annotation_fill_area_occurrence | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANNOTATION_FILL_AREA') IN TYPEOF(afao.item))) )) = 0);
    WR2: (SIZEOF(QUERY ( afao <* annotation_fill_area_occurrence | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CARTESIAN_POINT') IN TYPEOF(afao.fill_style_target))) )) = 0);
    WR3: (SIZEOF(QUERY ( afao <* annotation_fill_area_occurrence | (
        SIZEOF(QUERY ( psa <* afao.styles | (SIZEOF(QUERY ( cs <* psa.
        styles | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CURVE_STYLE') IN TYPEOF(cs)) )) = 0) )) = 0) )) = 0);

END_RULE; -- fill_area_occurrence_constraint

```

```

RULE fill_area_style_constraint FOR (fill_area_style);

```

```

WHERE
    WR1: (SIZEOF(QUERY ( fas <* fill_area_style | (NOT (SIZEOF(fas.
        fill_styles) = 1)) )) = 0);

END_RULE; -- fill_area_style_constraint

```

```

RULE fill_area_style_unique_constraint FOR (fill_area_style);

```

```

LOCAL
    name_bag : BAG OF STRING := [];
    r        : BAG OF fill_area_style := QUERY ( r <* fill_area_style |
        TRUE );
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].name) THEN

```

```

        IF r[i]\fill_area_style.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + r[i].name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- fill_area_style_unique_constraint

RULE functional_unit_definition_unique_constraint FOR (functional_unit);

LOCAL
    name_bag : BAG OF STRING := [];
    fud      : BAG OF functional_unit := QUERY ( fu <* functional_unit
        | (fu.frame_of_reference.name IN [
            'functional network design', 'functional design usage']) );
    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(fud) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF EXISTS(fud[i].formation.of_product.name) THEN
        IF fud[i].formation.of_product.name IN name_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            name_bag := name_bag + fud[i].formation.of_product.name;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- functional_unit_definition_unique_constraint

RULE functional_unit_network_node_definition_unique_constraint FOR (
    network_node_definition);

LOCAL
    fu_bag    : BAG OF functional_unit;
    pdr_bag   : BAG OF product_definition_relationship;
    name_bag  : BAG OF STRING := [];
    nnd_bag   : BAG OF network_node_definition;

```

```

    funnd      : BAG OF network_node_definition := QUERY ( nnd <*
        network_node_definition | (nnd.frame_of_reference.name =
            'functional network design') );
    pass       : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(funnd) BY 1;
    IF EXISTS(funnd[i].formation.of_product.name) THEN
        IF NOT (funnd[i].formation.of_product.name IN name_bag) THEN
            name_bag := name_bag + funnd[i].formation.of_product.name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    fu_bag := [];
    nnd_bag := QUERY ( nnd <* funnd | (nnd.formation.of_product.name =
        name_bag[i]) );
    REPEAT j := 1 TO SIZEOF(nnd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(nnd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'FUNCTIONAL_UNIT') IN TYPEOF(pdr.relating_product_definition))
            AND (pdr.name = 'functional unit network node')) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].relating_product_definition) THEN
                IF pdr_bag[k].relating_product_definition IN fu_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    fu_bag := fu_bag + pdr_bag[k].relating_product_definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- functional_unit_network_node_definition_unique_constraint

RULE functional_unit_network_terminal_definition_bus_assignment_constraint
    FOR ( property_definition_relationship);

WHERE

```

```

WR1: (SIZEOF(QUERY ( pdr <* property_definition_relationship | ((pdr.
    name =
    'functional unit network terminal definition bus assignment')
    AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(pdr.
    relating_property_definition.definition)))) )) = 0);
WR2: (SIZEOF(QUERY ( pdr <* property_definition_relationship | ((pdr.
    name =
    'functional unit network terminal definition bus assignment')
    AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(pdr.
    related_property_definition.definition)) AND (SIZEOF(
    QUERY ( futd <* USEDIN(pdr.related_property_definition.
    definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT_TERMINAL_DEFINITION.OF_SHAPE') IN TYPEOF(futd)) ))
    > 0)))) )) = 0);

END_RULE;

-- functional_unit_network_terminal_definition_bus_assignment_constraint

RULE functional_unit_terminal_bus_assignment_constraint FOR (
    property_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* property_definition_relationship | ((pdr.
        name = 'functional unit terminal bus assignment') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BUS_STRUCTURAL_DEFINITION') IN TYPEOF(pdr.
        relating_property_definition.definition)))) )) = 0);

END_RULE; -- functional_unit_terminal_bus_assignment_constraint

RULE functional_unit_unique_constraint FOR (component_functional_unit);

LOCAL
    cfu_bag : BAG OF component_functional_unit;
    fu_bag  : BAG OF functional_unit;
    pdr_bag : BAG OF product_definition_relationship;
    fu      : BAG OF component_functional_unit := QUERY ( cfu <*
        component_functional_unit | (cfu.frame_of_reference.name =
        'functional occurrence') );
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(fu) BY 1;

```



```

    IF EXISTS(fu[i].id) THEN
        IF NOT (fu[i].id IN id_bag) THEN
            id_bag := id_bag + fu[i].id;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(id_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    fu_bag := [];
    cfu_bag := QUERY ( cfu <* fu | (cfu.id = id_bag[i]) );
    REPEAT j := 1 TO SIZEOF(cfu_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(cfu_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | ((pdr.name = 'network composition') AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'FUNCTIONAL_UNIT') IN TYPEOF(pdr.relatering_product_definition))
            AND (pdr.relatering_product_definition.frame_of_reference.name =
            'functional network design')) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].relatering_product_definition) THEN
                IF pdr_bag[k].relatering_product_definition IN fu_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    fu_bag := fu_bag + pdr_bag[k].relatering_product_definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- functional_unit_unique_constraint

RULE functional_unit_usage_view_terminal_definition_unique_constraint FOR (
    functional_unit_terminal_definition);

LOCAL
    fu_bag    : BAG OF functional_unit;
    futd_bag  : BAG OF functional_unit_terminal_definition;
    name_bag  : BAG OF STRING := [];
    adr_bag   : BAG OF applied_document_reference;
    pass      : BOOLEAN := TRUE;

```

```

END_LOCAL;
REPEAT i := 1 TO SIZEOF(functional_unit_terminal_definition) BY 1;
  IF EXISTS(functional_unit_terminal_definition[i].name) THEN
    IF NOT (functional_unit_terminal_definition[i].name IN name_bag)
      THEN
        name_bag := name_bag + functional_unit_terminal_definition[i].
          name;
      END_IF;
    END_IF;
  END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  fu_bag := [];
  futd_bag := QUERY ( futd <* functional_unit_terminal_definition | ((
    futd.name = name_bag[i]) AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT') IN TYPEOF(futd.of_shape.definition)) AND (futd
        .of_shape.definition.frame_of_reference.name =
        'functional design usage')) );
  REPEAT j := 1 TO SIZEOF(futd_bag) BY 1;
    IF EXISTS(futd_bag[j].of_shape.definition) THEN
      IF futd_bag[j].of_shape.definition IN fu_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        fu_bag := fu_bag + futd_bag[j].of_shape.definition;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- functional_unit_usage_view_terminal_definition_unique_constraint

```

```

RULE functional_usage_view_to_part_terminal_assignment_constraint FOR (
  shape_aspect_relationship);

```

```

WHERE
  WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'functional terminal allocation') AND (NOT (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'FUNCTIONAL_UNIT_TERMINAL_DEFINITION') IN TYPEOF(sar.
        relating_shape_aspect)) AND (sar.relatng_shape_aspect.
        description = 'scalar terminal')))) ) = 0);
  WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'functional terminal allocation') AND (NOT (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'ASSEMBLY_MODULE_TERMINAL') IN TYPEOF(sar.related_shape_aspect))
    OR (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'BARE_DIE_TERMINAL') IN TYPEOF(sar.related_shape_aspect)) OR
    (( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERCONNECT_MODULE_TERMINAL') IN TYPEOF(sar.
    related_shape_aspect)) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGE_TERMINAL') IN TYPEOF(sar.related_shape_aspect)) OR ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_PART_TERMINAL') IN TYPEOF(sar.related_shape_aspect)))))) ))
    = 0);

```

```
END_RULE; -- functional_usage_view_to_part_terminal_assignment_constraint
```

```

RULE functional_usage_view_to_part_terminal_assignment_unique_constraint
    FOR ( shape_aspect_relationship);

```

```

LOCAL
    futd_bag : BAG OF functional_unit_terminal_definition := [];
    sar_bag : BAG OF shape_aspect_relationship;
    fuvtpa : BAG OF shape_aspect_relationship := QUERY ( r <*
        shape_aspect_relationship | (r.name =
        'functional terminal allocation') );
    pass : BOOLEAN := TRUE;
    sa_bag : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(fuvtpa) BY 1;
    IF EXISTS(fuvtpa[i].relating_shape_aspect) THEN
        IF fuvtpa[i].relating_shape_aspect.description =
            'scalar terminal' THEN
            IF NOT (fuvtpa[i].relating_shape_aspect IN futd_bag) THEN
                futd_bag := futd_bag + fuvtpa[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(futd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sar_bag := QUERY ( sar <* fuvtpa | ((sar.relating_shape_aspect :=:
        futd_bag[i]) AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ASSEMBLY_MODULE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PACKAGED_PART_TERMINAL'] * TYPEOF(sar.related_shape_aspect)) > 0)) );
sa_bag := [];
REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF EXISTS(sar_bag[j].related_shape_aspect) THEN
        IF sar_bag[j].related_shape_aspect IN sa_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            sa_bag := sa_bag + sar_bag[j].related_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE;

-- functional_usage_view_to_part_terminal_assignment_unique_constraint

RULE
functional_usage_view_to_printed_part_template_terminal_assignment_constraint
FOR ( shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'functional terminal allocation') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT_TERMINAL_DEFINITION') IN TYPEOF(sar.
        relating_shape_aspect)) AND (sar.relate_shape_aspect.
        description = 'scalar terminal')))) )) = 0);
    WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'functional terminal allocation') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(sar.
        related_shape_aspect)) AND ((sar.related_shape_aspect.
        description = 'interface terminal') OR (sar.
        related_shape_aspect.description = 'join terminal')))) )) = 0);

END_RULE;

-- functional_usage_view_to_printed_part_template_terminal_assignment_constraint

RULE
functional_usage_view_to_printed_part_template_terminal_assignment_unique_constraint
FOR ( shape_aspect_relationship);

LOCAL

```

```

futd_bag : BAG OF functional_unit_terminal_definition := [];
sar_bag : BAG OF shape_aspect_relationship;
pptt_bag : BAG OF printed_part_template_terminal;
pass : BOOLEAN := TRUE;
fuvtpptta : BAG OF shape_aspect_relationship := QUERY ( r <*
    shape_aspect_relationship | (r.name =
    'functional terminal allocation') );
END_LOCAL;
REPEAT i := 1 TO SIZEOF(fuvtpptta) BY 1;
    IF EXISTS(fuvtpptta[i].relating_shape_aspect) THEN
        IF fuvtpptta[i].relating_shape_aspect.description =
            'scalar terminal' THEN
            IF NOT (fuvtpptta[i].relating_shape_aspect IN futd_bag) THEN
                futd_bag := futd_bag + fuvtpptta[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(futd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pptt_bag := [];
    sar_bag := QUERY ( sar <* fuvtpptta | ((sar.relating_shape_aspect
        := futd_bag[i]) AND (sar.related_shape_aspect.description IN [
        'interface terminal', 'join terminal']) AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(sar.
        related_shape_aspect))) );
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF sar_bag[j].related_shape_aspect IN pptt_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                pptt_bag := pptt_bag + sar_bag[j].related_shape_aspect;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE;

-- functional_usage_view_to_printed_part_template_terminal_assignment_unique_constraint
RULE geometric_tolerance_qualifier_constraint FOR (measure_qualification);

```

WHERE

```
WR1: (SIZEOF(QUERY ( mq <* measure_qualification | (((mq.name =
    'predefined geometric tolerance qualifier') OR (mq.name =
    'user defined geometric tolerance qualifier')) AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_WITH_UNIT') IN TYPEOF(mq.qualified_measure)) AND (
    SIZEOF(QUERY ( pugt <* USEDIN(mq.qualified_measure,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE.MAGNITUDE') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE') IN TYPEOF(pugt)) )) = 0)))))) =
    0);
```

END_RULE; -- geometric_tolerance_qualifier_constraint

RULE inter_stratum_extent_constraint FOR (
 product_definition_relationship);

WHERE

```
WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
    name = 'inter stratum extent') AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM') IN TYPEOF(pdr.related_product_definition)) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM') IN TYPEOF(pdr.relatng_product_definition)))))) =
    0);
```

END_RULE; -- inter_stratum_extent_constraint

RULE interconnect_component_interface_terminal_unique_constraint FOR (
 component_interface_terminal);

LOCAL

```
cit_bag : BAG OF component_interface_terminal;
name_bag : BAG OF STRING;
pass : BOOLEAN := TRUE;
icd_bag : BAG OF interface_component_definition := [];
icit : BAG OF component_interface_terminal := QUERY ( cit <*
    component_interface_terminal | (cit.description =
    'interconnect component interface terminal') );
```

END_LOCAL;

REPEAT i := 1 TO SIZEOF(icit) BY 1;

IF EXISTS(icit[i].of_shape.definition) THEN

```
IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERFACE_COMPONENT_DEFINITION') IN TYPEOF(icit[i].of_shape.
    definition) THEN
```

IF NOT (icit[i].of_shape.definition IN icd_bag) THEN

```
icd_bag := icd_bag + icit[i].of_shape.definition;
```

END_IF;

```

        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(icd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    name_bag := [];
    cit_bag := QUERY ( cit <* icit | (cit.of_shape.definition :=:
        icd_bag[i]) );
    REPEAT j := 1 TO SIZEOF(cit_bag) BY 1;
        IF EXISTS(cit_bag[j].name) THEN
            IF cit_bag[j].name IN name_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                name_bag := name_bag + cit_bag[j].name;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- interconnect_component_interface_terminal_unique_constraint

RULE interconnect_component_join_terminal_unique_constraint FOR (
    component_terminal);

LOCAL
    name_bag : BAG OF STRING;
    cd_bag   : BAG OF component_definition := [];
    icjt     : BAG OF component_terminal := QUERY ( ct <*
        component_terminal | (ct.description =
            'interconnect component join terminal') );
    ct_bag   : BAG OF component_terminal;
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(icjt) BY 1;
    IF EXISTS(icjt[i].of_shape.definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'COMPONENT_DEFINITION') IN TYPEOF(icjt[i].of_shape.definition)
            THEN
            IF NOT (icjt[i].of_shape.definition IN cd_bag) THEN
                cd_bag := cd_bag + icjt[i].of_shape.definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(cd_bag) BY 1;

```

```

    IF NOT pass THEN
        ESCAPE;
    END_IF;
    name_bag := [];
    ct_bag := QUERY ( ct <* icjt | (ct.of_shape.definition :=: cd_bag[i]) );
    REPEAT j := 1 TO SIZEOF(ct_bag) BY 1;
        IF EXISTS(ct_bag[j].name) THEN
            IF ct_bag[j].name IN name_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                name_bag := name_bag + ct_bag[j].name;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- interconnect_component_join_terminal_unique_constraint

RULE interconnect_module_stratum_assembly_relationship_constraint FOR (
    assembly_component_usage);

WHERE
    WR1: (SIZEOF(QUERY ( acu <* assembly_component_usage | ((acu.name =
        'interconnect module stratum assembly relationship') AND (NOT
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT') IN TYPEOF(acu.relatng_product_definition)))) )
        = 0);
    WR2: (SIZEOF(QUERY ( acu <* assembly_component_usage | ((acu.name =
        'interconnect module stratum assembly relationship') AND (NOT
        (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM') IN TYPEOF(acu.related_product_definition)))) )
        = 0);

END_RULE; -- interconnect_module_stratum_assembly_relationship_constraint

RULE interconnect_module_stratum_assembly_relationship_unique_constraint
    FOR ( assembly_component_usage);

LOCAL
    acu_bag      : BAG OF assembly_component_usage;
    refdes_bag    : BAG OF STRING;
    pu_bag        : BAG OF physical_unit := [];
    imsar         : BAG OF assembly_component_usage := QUERY ( acu <*
        assembly_component_usage | (acu.name =
        'interconnect module stratum assembly relationship') );
    pass          : BOOLEAN := TRUE;
END_LOCAL;

```



```

REPEAT i := 1 TO SIZEOF(imsar) BY 1;
  IF EXISTS(imsar[i].relating_product_definition) THEN
    IF (imsar[i].relating_product_definition.frame_of_reference.name =
      'physical design') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT') IN TYPEOF(imsar[i].relating_product_definition))
    THEN
      IF NOT (imsar[i].relating_product_definition IN pu_bag) THEN
        pu_bag := pu_bag + imsar[i].relating_product_definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  refdes_bag := [];
  acu_bag := QUERY ( acu <* imsar | (acu.relating_product_definition
    :=: pu_bag[i]) );
  REPEAT j := 1 TO SIZEOF(acu_bag) BY 1;
    IF EXISTS(acu_bag[j].reference_designator) THEN
      IF acu_bag[j].reference_designator IN refdes_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        refdes_bag := refdes_bag + acu_bag[j].reference_designator;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE;

```

```
-- interconnect_module_stratum_assembly_relationship_unique_constraint
```

```
RULE interconnect_module_surface_feature_constraint FOR (shape_aspect);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
    'interconnect module surface feature') AND (NOT ((SIZEOF(
    QUERY ( prpc <* USEDIN(sa.of_shape.definition\
    product_definition.formation.of_product,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | (prpc\
    product_category.name = 'interconnect module') )) >= 1) AND (
    sa.of_shape.definition.frame_of_reference.name =
    'physical design usage') AND (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PHYSICAL_UNIT') IN TYPEOF(sa.of_shape.definition)) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PHYSICAL_UNIT') IN TYPEOF(sa.of_shape.
        definition)) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PHYSICAL_UNIT') IN TYPEOF(sa.of_shape.
        definition)))))) = 0);

END_RULE; -- interconnect_module_surface_feature_constraint

RULE interface_component_allocation_constraint FOR (
    product_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name = 'interface component allocation') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(pdr.
        relating_product_definition)) AND (pdr.
        relating_product_definition.description = 'mating connector')))) ))
        = 0);

END_RULE; -- interface_component_allocation_constraint

RULE internal_stratum_access_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    sar_bag : BAG OF shape_aspect_relationship;
    isa      : BAG OF shape_aspect_relationship := QUERY ( r <*
        shape_aspect_relationship | (r.name =
        'internal stratum access') );
    paa_bag : BAG OF probe_access_area := [];
    pass    : BOOLEAN := TRUE;
    pp_bag  : BAG OF plated_passage;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(isa) BY 1;
    IF EXISTS(isa[i].relating_shape_aspect) THEN
        IF (isa[i].relating_shape_aspect.description =
            'internal probe access area') AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROBE_ACCESS_AREA') IN TYPEOF(isa[i].relating_shape_aspect))
            THEN
            IF NOT (isa[i].relating_shape_aspect IN paa_bag) THEN
                paa_bag := paa_bag + isa[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(paa_bag) BY 1;

```

```

IF NOT pass THEN
    ESCAPE;
END_IF;
pp_bag := [];
sar_bag := QUERY ( sar <* isa | ((sar.relateing_shape_aspect :=:
    paa_bag[i]) AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLATED_PASSAGE') IN TYPEOF(sar.related_shape_aspect)) AND (sar.
        related_shape_aspect.description IN [
            'bonded conductive base blind via', 'buried via',
            'interfacial connection', 'non conductive base blind via',
            'plated conductive base blind via']))) );
REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].related_shape_aspect) THEN
        IF sar_bag[j].related_shape_aspect IN pp_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            pp_bag := pp_bag + sar_bag[j].related_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- internal_stratum_access_unique_constraint

RULE junction_vertex_allocation_constraint FOR (
    shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'junction vertex allocation') AND (NOT (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LAYER_CONNECTION_POINT') IN TYPEOF(sar.related_shape_aspect))
            AND (sar.related_shape_aspect.description =
                'explicitly located')) OR (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'JOIN_SHAPE_ASPECT') IN TYPEOF(sar.related_shape_aspect)) AND
            (sar.related_shape_aspect.name = 'inter stratum join'))))) ))
        = 0);
    WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'junction vertex allocation') AND (NOT (sar.
            relateing_shape_aspect.description = 'topological junction')) ))
        = 0);

END_RULE; -- junction_vertex_allocation_constraint

```

```

RULE junction_vertex_allocation_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    jva      : BAG OF shape_aspect_relationship := QUERY ( sar <*
        shape_aspect_relationship | (sar.name =
            'junction vertex allocation') );
    pass      : BOOLEAN := TRUE;
    sa_bag    : BAG OF shape_aspect := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(jva) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF EXISTS(jva[i].relating_shape_aspect) THEN
        IF jva[i].relating_shape_aspect.description =
            'topological junction' THEN
            IF jva[i].relating_shape_aspect IN sa_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                sa_bag := sa_bag + jva[i].relating_shape_aspect;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- junction_vertex_allocation_unique_constraint

```

```

RULE land_physical_template_unique_constraint FOR (
    land_physical_template);

LOCAL
    duc_bag  : BAG OF document_usage_constraint := [];
    adr_bag  : BAG OF applied_document_reference;
    pass     : BOOLEAN := TRUE;
    duc      : BAG OF document_usage_constraint;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(land_physical_template) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    adr_bag := USEDIN(land_physical_template[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS');
    REPEAT j := 1 TO SIZEOF(adr_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```

END_IF;
duc := QUERY ( r <* USEDIN(adr_bag[j].assigned_document,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (r.subject_element =
'predefined classification code') );
REPEAT k := 1 TO SIZEOF(duc) BY 1;
  IF EXISTS(duc[k]) THEN
    IF duc[k] IN duc_bag THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      duc_bag := duc_bag + duc[k];
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
WHERE
  WR1: pass;

END_RULE; -- land_physical_template_unique_constraint

```

```

RULE land_template_terminal_unique_constraint FOR (
  land_template_terminal);

LOCAL
  ltt      : BAG OF land_template_terminal := QUERY ( sa <*
    land_template_terminal | (sa.description IN [
      'interface terminal', 'join terminal']) );
  lpt_bag  : BAG OF land_physical_template;
  name_bag : BAG OF STRING := [];
  sar_bag  : BAG OF shape_aspect_relationship;
  ltt_bag  : BAG OF land_template_terminal;
  pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ltt) BY 1;
  IF EXISTS(ltt[i].name) THEN
    IF NOT (ltt[i].name IN name_bag) THEN
      name_bag := name_bag + ltt[i].name;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  lpt_bag := [];
  ltt_bag := QUERY ( sa <* ltt | (sa.name = name_bag[i]) );
  REPEAT j := 1 TO SIZEOF(ltt_bag) BY 1;
    IF NOT pass THEN

```

```

        ESCAPE;
    END_IF;
    sar_bag := QUERY ( sar <* USEDIN(ltt_bag[j],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
        = 'associated definition') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LAND_PHYSICAL_TEMPLATE') IN TYPEOF(sar.relater_shape_aspect)))) );
    REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[k].relater_shape_aspect) THEN
            IF sar_bag[k].relater_shape_aspect IN lpt_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                lpt_bag := lpt_bag + sar_bag[k].relater_shape_aspect;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- land_template_terminal_unique_constraint

```

```

RULE layout_spacing_requirement_non_conforming_design_object_constraint
    FOR ( shape_aspect_relationship);

```

```

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'layout spacing requirement violation') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRUTUM_FEATURE'] * TYPEOF(sar.related_shape_aspect)) = 0)) ))
        = 0);
    WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'layout spacing requirement violation') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_INTERFACE_TERMINAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRUTUM_FEATURE'] * TYPEOF(sar.relater_shape_aspect)) = 0)) ))
        = 0);

```

```

    = 0);
WR3: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'layout spacing requirement violation') AND (NOT (SIZEOF(
    QUERY ( grp <* USEDIN(sar,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP') IN TYPEOF(grp)) AND (grp.name =
    'layout spacing requirements property')) )) = 1))) )) = 0);

END_RULE;

-- layout_spacing_requirement_non_conforming_design_object_constraint

RULE layout_spacing_requirement_occurrence_unique_constraint FOR (
    grouped_requirements_property, characterized_object);

LOCAL
    pd_bag      : BAG OF property_definition;
    rga_bag      : BAG OF group_assignment;
    rdname_bag   : BAG OF STRING;
    ga_bag       : BAG OF group_assignment;
    ddo          : BAG OF characterized_object := QUERY ( co <*
        characterized_object | (co.description =
        'dependent design object category') );
    grp_bag      : BAG OF grouped_requirements_property;
    lsro         : BAG OF grouped_requirements_property := QUERY ( grp <*
        grouped_requirements_property | (grp\group.name =
        'layout spacing requirements property') );
    pass         : BOOLEAN := TRUE;
    co_bag       : BAG OF characterized_object;
    ddname_bag   : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(lsro) BY 1;
    ga_bag := USEDIN(lsro[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP');
    REPEAT j := 1 TO SIZEOF(ga_bag) BY 1;
        REPEAT k := 1 TO SIZEOF(ga_bag[j].items) BY 1;
            IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CHARACTERIZED_OBJECT') IN TYPEOF(ga_bag[j].items[k].
                definition)) AND (ga_bag[j].items[k].definition.description =
                'dependent design object category') THEN
                IF EXISTS(ga_bag[j].items[k].definition.name) THEN
                    IF NOT (ga_bag[j].items[k].definition.name IN ddname_bag)
                        THEN
                        ddname_bag := ddname_bag + ga_bag[j].items[k].definition.
                            name;
                    END_IF;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(ddname_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  rdname_bag := [];
  co_bag := QUERY ( co <* ddo | (co.name = ddname_bag[i]) );
  REPEAT j := 1 TO SIZEOF(co_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pd_bag := USEDIN(co_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION');
    REPEAT k := 1 TO SIZEOF(pd_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      ga_bag := QUERY ( ga <* USEDIN(pd_bag[k],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_GROUP_ASSIGNMENT.ITEMS') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUPED_REQUIREMENTS_PROPERTY') IN TYPEOF(ga.assigned_group)) );
      REPEAT l := 1 TO SIZEOF(ga_bag) BY 1;
        IF NOT pass THEN
          ESCAPE;
        END_IF;
        IF NOT (ga_bag[l].assigned_group IN grp_bag) THEN
          grp_bag := grp_bag + ga_bag[l].assigned_group;
        END_IF;
      END_REPEAT;
    REPEAT l := 1 TO SIZEOF(grp_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      rga_bag := USEDIN(grp_bag[l],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP');
      REPEAT m := 1 TO SIZEOF(rga_bag) BY 1;
        IF NOT pass THEN
          ESCAPE;
        END_IF;
        REPEAT n := 1 TO SIZEOF(rga_bag[m].items) BY 1;
          IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'CHARACTERIZED_OBJECT') IN TYPEOF(rga_bag[m].items[n]
              .definition)) AND (rga_bag[m].items[n].definition.
                description = 'reference design object category') THEN
            IF EXISTS(rga_bag[m].items[n].definition.name) THEN
              IF rga_bag[m].items[n].definition.name IN rdname_bag
                THEN
                  pass := FALSE;
                END_IF;
            END_IF;
          END_REPEAT;
        END_REPEAT;
      END_REPEAT;
    END_REPEAT;
  END_REPEAT;

```



```

        ESCAPE;
      ELSE
        rdname_bag := rdname_bag + rga_bag[m].items[n].
          definition.name;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- layout_spacing_requirement_occurrence_unique_constraint

RULE located_interconnect_module_thickness_requirement_unique_constraint
FOR ( property_definition_representation);

LOCAL
  rr_bag : BAG OF representation_relationship;
  pdr_bag : BAG OF property_definition_representation;
  ri_bag : BAG OF representation_item;
  pu_bag : BAG OF physical_unit := [];
  limrt : BAG OF property_definition_representation :=
    QUERY ( pdr <* property_definition_representation | (pdr.
      definition\property_definition.name =
        'located interconnect module thickness') );
  pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(limrt) BY 1;
  IF EXISTS(limrt[i].definition.definition.of_shape.definition) THEN
    IF NOT (limrt[i].definition.definition.of_shape.definition IN
      pu_bag) THEN
      pu_bag := pu_bag + limrt[i].definition.definition.of_shape.
        definition;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  ri_bag := [];
  pdr_bag := QUERY ( pdr <* limrt | (pdr.definition.definition.
    of_shape.definition :=: pu_bag[i]) );
  REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF NOT pass THEN

```

```

        ESCAPE;
    END_IF;
    rr_bag := QUERY ( rr <* USEDIN(pdr_bag[j].used_representation,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') | (rr\
        representation_relationship.name = 'reference location') );
    REPEAT k := 1 TO SIZEOF(rr_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        REPEAT l := 1 TO SIZEOF(rr_bag[k].rep_1.items) BY 1;
            IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                + 'AXIS2_PLACEMENT_2D') IN TYPEOF(rr_bag[k].rep_1.items[l]))
            OR ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'AXIS2_PLACEMENT_3D') IN TYPEOF(rr_bag[k].rep_1.items[l]))
            THEN
                IF EXISTS(rr_bag[k].rep_1.items[l]) THEN
                    IF rr_bag[k].rep_1.items[l] IN ri_bag THEN
                        pass := FALSE;
                        ESCAPE;
                    ELSE
                        ri_bag := ri_bag + rr_bag[k].rep_1.items[l];
                    END_IF;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE;

-- located_interconnect_module_thickness_requirement_unique_constraint

RULE mandatory_product_related_product_category_value FOR (
    product_related_product_category, product);

LOCAL
    coined                : BOOLEAN := FALSE;
    molded                 : BOOLEAN := FALSE;
    detail                 : BOOLEAN := FALSE;
    domain_thing           : BOOLEAN := FALSE;
    prpc                   : BAG OF
        product_related_product_category :=
        [];
    package                : BOOLEAN := FALSE;
    tool                   : BOOLEAN := FALSE;
    machined               : BOOLEAN := FALSE;

```

```

inseparable_assembly      : BOOLEAN := FALSE;
raw_material               : BOOLEAN := FALSE;
design_layer                : BOOLEAN := FALSE;
interconnect_module       : BOOLEAN := FALSE;
rolled                     : BOOLEAN := FALSE;
bare_die                   : BOOLEAN := FALSE;
formed                     : BOOLEAN := FALSE;
assembly_module            : BOOLEAN := FALSE;
document                   : BOOLEAN := FALSE;
customer_furnished_equipment : BOOLEAN := FALSE;
assembly                   : BOOLEAN := FALSE;
printed_part               : BOOLEAN := FALSE;
standard                   : BOOLEAN := FALSE;
generic_thing              : BOOLEAN := FALSE;
functionality              : BOOLEAN := FALSE;
piece_part                 : BOOLEAN := FALSE;
material                   : BOOLEAN := FALSE;
extruded                   : BOOLEAN := FALSE;
part                       : BOOLEAN := FALSE;
documentation_layer        : BOOLEAN := FALSE;
template_model             : BOOLEAN := FALSE;
requirements_model         : BOOLEAN := FALSE;
simulation_model           : BOOLEAN := FALSE;
pass                       : BOOLEAN := TRUE;
drawn                      : BOOLEAN := FALSE;
sheared                    : BOOLEAN := FALSE;
cast                       : BOOLEAN := FALSE;
packaged_part              : BOOLEAN := FALSE;
forged                     : BOOLEAN := FALSE;
interconnect               : BOOLEAN := FALSE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(product) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  prpc := USEDIN(product[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS');
  IF SIZEOF(prpc) = 1 THEN
    ESCAPE;
  ELSE
    IF SIZEOF(prpc) > 1 THEN
      bare_die := SIZEOF(QUERY ( p <* prpc | (p.name = 'bare die') ))
        = 1;
      design_layer := SIZEOF(QUERY ( p <* prpc | (p.name =
        'design layer') )) = 1;
      documentation_layer := SIZEOF(QUERY ( p <* prpc | (p.name =
        'documentation layer') )) = 1;
      functionality := SIZEOF(QUERY ( p <* prpc | (p.name =
        'functionality') )) = 1;
      interconnect := SIZEOF(QUERY ( p <* prpc | (p.name =
        'interconnect') )) = 1;
    END_IF;
  END_IF;
END_REPEAT;

```

```

interconnect_module := SIZEOF(QUERY ( p <* prpc | (p.name =
    'interconnect module') )) = 1;
package := SIZEOF(QUERY ( p <* prpc | (p.name = 'package') )) =
    1;
packaged_part := SIZEOF(QUERY ( p <* prpc | (p.name =
    'packaged part') )) = 1;
printed_part := SIZEOF(QUERY ( p <* prpc | (p.name =
    'printed part') )) = 1;
requirements_model := SIZEOF(QUERY ( p <* prpc | (p.name =
    'requirements model') )) = 1;
simulation_model := SIZEOF(QUERY ( p <* prpc | (p.name =
    'simulation model') )) = 1;
template_model := SIZEOF(QUERY ( p <* prpc | (p.name =
    'template model') )) = 1;
raw_material := SIZEOF(QUERY ( p <* prpc | (p.name =
    'raw material') )) = 1;
cast := SIZEOF(QUERY ( p <* prpc | (p.name = 'cast') )) = 1;
coined := SIZEOF(QUERY ( p <* prpc | (p.name = 'coined') )) = 1;
drawn := SIZEOF(QUERY ( p <* prpc | (p.name = 'drawn') )) = 1;
extruded := SIZEOF(QUERY ( p <* prpc | (p.name = 'extruded') ))
    = 1;
forged := SIZEOF(QUERY ( p <* prpc | (p.name = 'forged') )) = 1;
formed := SIZEOF(QUERY ( p <* prpc | (p.name = 'formed') )) = 1;
machined := SIZEOF(QUERY ( p <* prpc | (p.name = 'machined') ))
    = 1;
molded := SIZEOF(QUERY ( p <* prpc | (p.name = 'molded') )) = 1;
piece_part := SIZEOF(QUERY ( p <* prpc | (p.name = 'piece part') ))
    = 1;
rolled := SIZEOF(QUERY ( p <* prpc | (p.name = 'rolled') )) = 1;
customer_furnished_equipment := SIZEOF(QUERY ( p <* prpc | (p.
    name = 'customer furnished equipment') )) = 1;
inseparable_assembly := SIZEOF(QUERY ( p <* prpc | (p.name =
    'inseparable assembly') )) = 1;
standard := SIZEOF(QUERY ( p <* prpc | (p.name = 'standard') ))
    = 1;
assembly := SIZEOF(QUERY ( p <* prpc | (p.name = 'assembly') ))
    = 1;
detail := SIZEOF(QUERY ( p <* prpc | (p.name = 'detail') )) = 1;
part := SIZEOF(QUERY ( p <* prpc | (p.name = 'part') )) = 1;
document := SIZEOF(QUERY ( p <* prpc | (p.name = 'document') ))
    = 1;
tool := SIZEOF(QUERY ( p <* prpc | (p.name = 'tool') )) = 1;
material := SIZEOF(QUERY ( p <* prpc | (p.name = 'material') ))
    = 1;
generic_thing := part OR tool OR material OR cast OR coined OR
    drawn OR extruded OR forged OR formed OR machined OR molded
    OR piece_part OR raw_material OR rolled OR
    customer_furnished_equipment OR inseparable_assembly OR
    sheared;
domain_thing := assembly_module OR bare_die OR design_layer OR
    documentation_layer OR functionality OR interconnect OR
    interconnect_module OR package OR packaged_part OR

```

```

        printed_part OR requirements_model OR simulation_model OR
        template_model;
    pass := (assembly XOR detail) OR (generic_thing XOR document) OR
        standard OR (domain_thing XOR document);
    END_IF;
    END_IF;
    END_REPEAT;

WHERE
    WR1: pass;
    WR2: (SIZEOF(QUERY ( prpc <* product_related_product_category | (NOT (
        prpc\product_category.name IN ['assembly','assembly module',
        'bare die','cast','coined','customer furnished equipment',
        'design layer','detail','documentation layer','drawn',
        'extruded','forged','formed','functionality',
        'inseparable assembly','interconnect','interconnect module',
        'machined','material','molded','package','packaged part',
        'part','piece part','printed part','raw material',
        'requirements model','rolled','sheared','simulation model',
        'standard','template model','tool']))) )) = 0);

END_RULE; -- mandatory_product_related_product_category_value

RULE marking_constraint FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name = 'marking')
        AND (SIZEOF(QUERY ( pdr <* USEDIN(rep,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT') IN TYPEOF(pdr.definition)) )) > 0) AND (NOT (
        SIZEOF(QUERY ( place <* rep.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLACEMENT') IN TYPEOF(place)) AND (place.name =
        'marking location')) )) = 1))) )) = 0);

END_RULE; -- marking_constraint

RULE material_assembly_relationship_constraint FOR (
    assembly_component_usage);

WHERE
    WR1: (SIZEOF(QUERY ( acu <* assembly_component_usage | ((acu.name =
        'material assembly relationship') AND (SIZEOF(USEDIN(acu.
        relating_product_definition,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) = 0)) )) = 0);
    WR2: (SIZEOF(QUERY ( acu <* assembly_component_usage | ((acu.name =
        'material assembly relationship') AND (SIZEOF(USEDIN(acu.

```

```

        related_product_definition,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MATERIAL_DESIGNATION.DEFINITIONS')) = 0)) )) = 0);

END_RULE; -- material_assembly_relationship_constraint

RULE material_assembly_relationship_unique_constraint FOR (
    assembly_component_usage);

LOCAL
    acu_bag : BAG OF assembly_component_usage;
    md       : BAG OF material_designation;
    mar      : BAG OF assembly_component_usage := QUERY ( acu <*
        assembly_component_usage | (acu.name =
            'material assembly relationship') );
    pass     : BOOLEAN := TRUE;
    id_bag   : BAG OF STRING := [];
    md_bag   : BAG OF material_designation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(mar) BY 1;
    IF EXISTS(mar[i].related_product_definition.formation.of_product.id)
    THEN
        IF NOT (mar[i].related_product_definition.formation.of_product.id
            IN id_bag) THEN
            id_bag := id_bag + mar[i].related_product_definition.formation.
                of_product.id;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(id_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    md_bag := [];
    acu_bag := QUERY ( acu <* mar | (acu.related_product_definition.
        formation.of_product.id = id_bag[i]) );
    REPEAT j := 1 TO SIZEOF(acu_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        md := USEDIN(acu_bag[j].relating_product_definition,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MATERIAL_DESIGNATION.DEFINITIONS');
        REPEAT k := 1 TO SIZEOF(md) BY 1;
            IF EXISTS(md[k]) THEN
                IF md[k] IN md_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    md_bag := md_bag + md[k];
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;

```

```

        END_IF;
        END_REPEAT;
        END_REPEAT;
        END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- material_assembly_relationship_unique_constraint

RULE mating_connector_placement_relationship_constraint FOR (
    shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'mating connector placement') AND (SIZEOF(QUERY ( p <* sr.
        items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PLACEMENT') IN TYPEOF(p)) AND (p.name = 'connector placement')) ))
        = 0)) )) = 0);

END_RULE; -- mating_connector_placement_relationship_constraint

RULE mating_connector_termination_constraint FOR (shape_aspect);

WHERE
    WR1: (SIZEOF(QUERY ( sa <* shape_aspect | ((sa.description =
        'mating connector termination') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_DEFINITION') IN TYPEOF(sa.of_shape)) AND (sa.
        of_shape.description = 'mating connector')))) )) = 0);

END_RULE; -- mating_connector_termination_constraint

RULE mating_connector_termination_unique_constraint FOR (shape_aspect);

LOCAL
    cd_bag : BAG OF component_definition := [];
    sar_bag : BAG OF shape_aspect_relationship;
    ppt_bag : BAG OF packaged_part_terminal;
    mct : BAG OF shape_aspect := QUERY ( sa <* shape_aspect | (sa.
        description = 'mating connector termination') );
    pass : BOOLEAN := TRUE;
    sa_bag : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(mct) BY 1;
    IF EXISTS(mct[i].of_shape.definition) THEN
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'COMPONENT_DEFINITION') IN TYPEOF(mct[i].of_shape.definition))
    AND (mct[i].of_shape.definition.description =
        'mating connector') THEN
    IF NOT (mct[i].of_shape.definition IN cd_bag) THEN
        cd_bag := cd_bag + mct[i].of_shape.definition;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(cd_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    ppt_bag := [];
    sa_bag := QUERY ( sa <* mct | (sa.of_shape.definition :=: cd_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        sar_bag := QUERY ( sar <* USEDIN(sa_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
            = 'instantiated terminal') AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART_TERMINAL') IN TYPEOF(sar.relate_shape_aspect)))) );
        REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
            IF EXISTS(sar_bag[k].relate_shape_aspect) THEN
                IF sar_bag[k].relate_shape_aspect IN ppt_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    ppt_bag := ppt_bag + sar_bag[k].relate_shape_aspect;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- mating_connector_termination_unique_constraint

RULE model_parameter_unique_constraint FOR (model_parameter);

LOCAL
    mp_bag : BAG OF model_parameter;
    name_bag : BAG OF STRING := [];
    adr_bag : BAG OF applied_document_reference;
    pass : BOOLEAN := TRUE;
    d_bag : BAG OF document;

```



```

END_LOCAL;
REPEAT i := 1 TO SIZEOF(model_parameter) BY 1;
  IF EXISTS(model_parameter[i].name) THEN
    IF NOT (model_parameter[i].name IN name_bag) THEN
      name_bag := name_bag + model_parameter[i].name;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  mp_bag := QUERY ( mp <* model_parameter | (mp.name = name_bag[i]) );
  d_bag := [];
  REPEAT j := 1 TO SIZEOF(mp_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    adr_bag := USEDIN(mp_bag[j],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'APPLIED_DOCUMENT_REFERENCE.ITEMS');
    REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
      IF EXISTS(adr_bag[k].assigned_document) THEN
        IF adr_bag[k].assigned_document IN d_bag THEN
          pass := FALSE;
          ESCAPE;
        ELSE
          d_bag := d_bag + adr_bag[k].assigned_document;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- model_parameter_unique_constraint

```

```

RULE next_higher_assembly_interface_requirement_constraint FOR (
  specified_higher_usage_occurrence);

```

```

WHERE
  WR1: (SIZEOF(QUERY ( shuo <* specified_higher_usage_occurrence | (NOT
    (SIZEOF(QUERY ( pd <* USEDIN(shuo,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
        QUERY ( pdr <* USEDIN(pd,
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'SHAPE_REPRESENTATION') IN TYPEOF(pdr.used_representation))
    AND (pdr.used_representation.name = '3d bound volume shape')) ))
    = 0)) )) = 0)) )) = 0);

```

```
END_RULE; -- next_higher_assembly_interface_requirement_constraint
```

```

RULE next_higher_assembly_relationship_unique_constraint FOR (
    next_assembly_usage_occurrence);

```

```

LOCAL
    refdes_bag : BAG OF STRING;
    pu_bag      : BAG OF physical_unit := [];
    nauo_bag    : BAG OF next_assembly_usage_occurrence;
    pass        : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(next_assembly_usage_occurrence) BY 1;
    IF EXISTS(next_assembly_usage_occurrence[i].
        relating_product_definition) THEN
        IF (next_assembly_usage_occurrence[i].relating_product_definition.
            frame_of_reference.name = 'physical design') AND ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'PHYSICAL_UNIT') IN TYPEOF(next_assembly_usage_occurrence[i].
                    relating_product_definition)) THEN
            IF NOT (next_assembly_usage_occurrence[i].
                relating_product_definition IN pu_bag) THEN
                pu_bag := pu_bag + next_assembly_usage_occurrence[i].
                    relating_product_definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    refdes_bag := [];
    nauo_bag := QUERY ( nauo <* next_assembly_usage_occurrence | (nauo.
        relating_product_definition == pu_bag[i]) );
    REPEAT j := 1 TO SIZEOF(nauo_bag) BY 1;
        IF EXISTS(nauo_bag[j].reference_designator) THEN
            IF nauo_bag[j].reference_designator IN refdes_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                refdes_bag := refdes_bag + nauo_bag[j].reference_designator;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```
WHERE
```

```

WR1: pass;

END_RULE; -- next_higher_assembly_relationship_unique_constraint

RULE no_shape_for_make_from FOR (make_from_usage_option);

WHERE
  WR1: (SIZEOF(QUERY ( mfuo <* make_from_usage_option | (NOT (SIZEOF(
    QUERY ( pd <* USEDIN(mfuo,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(pd)) )) = 0)) )) = 0);

END_RULE; -- no_shape_for_make_from

RULE operational_requirement_occurrence_relationship_constraint FOR (
  property_definition_relationship);

WHERE
  WR1: (SIZEOF(QUERY ( pdr <* property_definition_relationship | (((pdr.
    name = 'and operation') OR (pdr.name = 'or operation')) AND (
    NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
    related_property_definition)))) )) = 0);
  WR2: (SIZEOF(QUERY ( pdr <* property_definition_relationship | (((pdr.
    name = 'and operation') OR (pdr.name = 'or operation')) AND (
    NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
    relating_property_definition)))) )) = 0);

END_RULE; -- operational_requirement_occurrence_relationship_constraint

RULE organization_unique_constraint FOR (organization);

LOCAL
  r      : BAG OF organization := QUERY ( r <* organization | TRUE );
  pass   : BOOLEAN := TRUE;
  id_bag : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
  IF EXISTS(r[i].id) THEN
    IF r[i]\organization.id IN id_bag THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      id_bag := id_bag + r[i].id;
    END_IF;
  END_REPEAT;

```

```

        END_IF;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- organization_unique_constraint

RULE package_external_reference_constraint FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'package external reference') AND (SIZEOF(QUERY ( dri <* rep.
        items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'design owner')))) )) = 1)) )) = 0);
    WR2: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'package external reference') AND (SIZEOF(QUERY ( dri <* rep.
        items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'part number')))) )) = 1)) )) = 0);
    WR3: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'package external reference') AND (SIZEOF(QUERY ( dri <* rep.
        items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'revision code')))) )) = 1)) )) = 0);
    WR4: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'package external reference') AND (SIZEOF(QUERY ( dri <* rep.
        items | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'product definition id')))) )) = 1)) )) = 0);

END_RULE; -- package_external_reference_constraint

RULE package_terminal_external_reference_constraint FOR (
    descriptive_representation_item);

WHERE
    WR1: (SIZEOF(QUERY ( dri <* descriptive_representation_item | ((dri.
        name = 'package terminal external reference') AND (SIZEOF(
        QUERY ( rep <* USEDIN(dri,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS') | (rep.name =
        'package external reference') )) = 0)) )) = 0);

END_RULE; -- package_terminal_external_reference_constraint

```

```

RULE package_terminal_external_reference_unique_constraint FOR (
    descriptive_representation_item);

LOCAL
    pter      : BAG OF descriptive_representation_item := QUERY ( dri <*
        descriptive_representation_item | (dri.name =
            'package terminal external reference') );
    desc_bag  : BAG OF STRING := [];
    dri_bag   : BAG OF descriptive_representation_item;
    rep_bag   : BAG OF representation;
    pass      : BOOLEAN := TRUE;
    r_bag     : BAG OF representation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pter) BY 1;
    IF EXISTS(pter[i].description) THEN
        IF NOT (pter[i].description IN desc_bag) THEN
            desc_bag := desc_bag + pter[i].description;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(desc_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    r_bag := [];
    dri_bag := QUERY ( dri <* pter | (dri.description = desc_bag[i]) );
    REPEAT j := 1 TO SIZEOF(dri_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        rep_bag := QUERY ( r <* USEDIN(dri_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION.ITEMS') | (r.name =
                'package external reference') );
        REPEAT k := 1 TO SIZEOF(rep_bag) BY 1;
            IF EXISTS(rep_bag[k]) THEN
                IF rep_bag[k] IN r_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    r_bag := r_bag + rep_bag[k];
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- package_terminal_external_reference_unique_constraint

```

```

RULE package_unique_constraint FOR (package, externally_defined_package,
    library_defined_package);

LOCAL
    edp      : BAG OF physical_unit := QUERY ( r <*
        externally_defined_package | (r.frame_of_reference.name =
            'physical design usage') );
    pdr_bag  : BAG OF property_definition_representation;
    pd_bag   : BAG OF property_definition;
    p        : BAG OF physical_unit := QUERY ( r <* package | (r.
        frame_of_reference.name = 'physical design usage') );
    pu       : BAG OF physical_unit;
    ldp      : BAG OF physical_unit := QUERY ( r <*
        library_defined_package | (r.frame_of_reference.name =
            'physical design usage') );
    pass     : BOOLEAN := TRUE;
    r_bag    : BAG OF representation := [];
END_LOCAL;
pu := p + edp + ldp;
REPEAT i := 1 TO SIZEOF(pu) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pd_bag := USEDIN(pu[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION');
    REPEAT j := 1 TO SIZEOF(pd_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(pd_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
            used_representation.name = 'registered case style') );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].used_representation) THEN
                IF pdr_bag[k].used_representation IN r_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    r_bag := r_bag + pdr_bag[k].used_representation;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- package_unique_constraint

```

```

RULE packaged_connector_terminal_relationship_unique_constraint FOR (
    packaged_connector_terminal_relationship);

LOCAL
    it_bag    : BAG OF packaged_part_terminal;
    pctr_bag   : BAG OF packaged_connector_terminal_relationship;
    pass1      : BOOLEAN := TRUE;
    pass2      : BOOLEAN := TRUE;
    pu_bag     : BAG OF physical_unit := [];
    jt_bag     : BAG OF packaged_part_terminal;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(packaged_connector_terminal_relationship) BY 1;
    IF EXISTS(packaged_connector_terminal_relationship[i].of_shape.
        definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT') IN TYPEOF(
                packaged_connector_terminal_relationship[i].of_shape.definition)
            THEN
                IF NOT (packaged_connector_terminal_relationship[i].of_shape.
                    definition IN pu_bag) THEN
                    pu_bag := pu_bag + packaged_connector_terminal_relationship[i]
                        .of_shape.definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
    IF NOT (pass1 OR pass2) THEN
        ESCAPE;
    END_IF;
    it_bag := [];
    jt_bag := [];
    pctr_bag := QUERY ( r <* packaged_connector_terminal_relationship |
        (r.of_shape.definition ==: pu_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pctr_bag) BY 1;
        IF NOT (pass1 OR pass2) THEN
            ESCAPE;
        END_IF;
        IF pass1 AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGED_PART_TERMINAL') IN TYPEOF(pctr_bag[j].
                relating_shape_aspect)) AND (pctr_bag[j].relating_shape_aspect.
                    description = 'join terminal') THEN
            IF EXISTS(pctr_bag[j].relating_shape_aspect) THEN
                IF pctr_bag[j].relating_shape_aspect IN jt_bag THEN
                    pass1 := FALSE;
                    ESCAPE;
                ELSE
                    jt_bag := jt_bag + pctr_bag[j].relating_shape_aspect;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;

```

```

END_IF;
IF pass2 AND ((
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PACKAGED_PART_TERMINAL') IN TYPEOF(pctr_bag[j].
  related_shape_aspect)) AND (pctr_bag[j].related_shape_aspect.
  description = 'interface terminal') THEN
  IF EXISTS(pctr_bag[j].related_shape_aspect) THEN
    IF pctr_bag[j].related_shape_aspect IN it_bag THEN
      pass2 := FALSE;
      ESCAPE;
    ELSE
      it_bag := it_bag + pctr_bag[j].related_shape_aspect;
    END_IF;
  END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass1;
  WR2: pass2;

END_RULE; -- packaged_connector_terminal_relationship_unique_constraint

RULE packaged_part_unique_constraint FOR (packaged_part,
  externally_defined_packaged_part,
  library_defined_packaged_part);

LOCAL
  p_bag   : BAG OF physical_unit;
  fu_bag   : BAG OF functional_unit := [];
  pdr_bag  : BAG OF product_definition_relationship;
  edpp     : BAG OF physical_unit := QUERY ( r <*
    externally_defined_packaged_part | (r.frame_of_reference.
    name = 'physical design usage') );
  aaa_bag  : BAG OF applied_action_assignment;
  ldpp     : BAG OF physical_unit := QUERY ( r <*
    library_defined_packaged_part | (r.frame_of_reference.name
    = 'physical design usage') );
  pp       : BAG OF physical_unit := QUERY ( r <* packaged_part | (r.
    frame_of_reference.name = 'physical design usage') );
  pu       : BAG OF physical_unit;
  pass     : BOOLEAN := TRUE;
  pp_bag   : BAG OF packaged_part;
END_LOCAL;
pu := pp + edpp + ldpp;
REPEAT i := 1 TO SIZEOF(pu) BY 1;
  pdr_bag := QUERY ( pdr <* USEDIN(pu[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') | (
    (pdr.name = 'implemented function') AND (pdr.

```



```

        relating_product_definition.frame_of_reference.name =
        'functional design usage') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'FUNCTIONAL_UNIT') IN TYPEOF(pdr.relatng_product_definition))) );
REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF EXISTS(pdr_bag[j].relating_product_definition) THEN
        IF NOT (pdr_bag[j].relating_product_definition IN fu_bag) THEN
            fu_bag := fu_bag + pdr_bag[j].relating_product_definition;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(fu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pp_bag := QUERY ( r <* pu | (SIZEOF(QUERY ( pdr <* USEDIN(fu_bag[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATNG_PRODUCT_DEFINITION') |
        ((pdr.name = 'implemented function') AND (pdr.
        relating_product_definition :=: r)) )) > 0) );
    p_bag := [];
    REPEAT j := 1 TO SIZEOF(pp_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        pdr_bag := QUERY ( pdr <* USEDIN(pp_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
            | ((pdr.name = 'used package') AND (pdr.
            relating_product_definition.frame_of_reference.name =
            'physical design usage') AND (((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PACKAGE') IN TYPEOF(pdr.relatng_product_definition)) OR ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'EXTERNALLY_DEFINED_PACKAGE') IN TYPEOF(pdr.
            relating_product_definition)) OR ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LIBRARY_DEFINED_PACKAGE') IN TYPEOF(pdr.
            relating_product_definition)))) );
        REPEAT k := 1 TO SIZEOF(pdr_bag) BY 1;
            IF EXISTS(pdr_bag[k].relating_product_definition) THEN
                IF pdr_bag[k].relating_product_definition IN p_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    p_bag := p_bag + pdr_bag[k].relating_product_definition;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    WR1: pass;

END_RULE; -- packaged_part_unique_constraint

RULE part_device_terminal_model_port_assignment_unique_constraint FOR (
    property_definition_representation);

LOCAL
    pdr_bag : BAG OF property_definition_representation;
    amp_bag : BAG OF analytical_model_port := [];
    pass    : BOOLEAN := TRUE;
    sa_bag  : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(property_definition_representation) BY 1;
    IF EXISTS(property_definition_representation[i].used_representation)
    THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ANALYTICAL_MODEL_PORT') IN TYPEOF(
                property_definition_representation[i].used_representation)
        THEN
            IF NOT (property_definition_representation[i].
                used_representation IN amp_bag) THEN
                amp_bag := amp_bag + property_definition_representation[i].
                    used_representation;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(amp_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sa_bag := [];
    pdr_bag := QUERY ( pdr <* property_definition_representation | (pdr.
        used_representation == amp_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
        IF EXISTS(pdr_bag[j].definition.definition) THEN
            IF pdr_bag[j].definition.definition IN sa_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                sa_bag := sa_bag + pdr_bag[j].definition.definition;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

```

```
END_RULE; -- part_device_terminal_model_port_assignment_unique_constraint
```

```
RULE part_device_terminal_unique_constraint FOR (shape_aspect);
```

```
LOCAL
  pdr_bag : BAG OF product_definition_relationship := [];
  pd_bag  : BAG OF property_definition;
  bdt_bag : BAG OF bare_die_terminal;
  sa      : BAG OF bare_die_terminal;
  pass    : BOOLEAN := TRUE;
  sa_bag  : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(shape_aspect) BY 1;
  IF EXISTS(shape_aspect[i].of_shape.definition) THEN
    IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PRODUCT_DEFINITION_RELATIONSHIP') IN TYPEOF(shape_aspect[i].
        of_shape.definition)) AND (shape_aspect[i].of_shape.definition.
          description = 'part device') THEN
      IF NOT (shape_aspect[i].of_shape.definition IN pdr_bag) THEN
        pdr_bag := pdr_bag + shape_aspect[i].of_shape.definition;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pdr_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  bdt_bag := [];
  sa_bag := QUERY ( sa <* shape_aspect | (sa.of_shape.definition ::=
    pdr_bag[i]) );
  REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pd_bag := QUERY ( pd <* USEDIN(sa_bag[j].of_shape.definition.
      related_product_definition,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF([
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'BARE_DIE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'EXTERNALLY_DEFINED_BARE_DIE',
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'LIBRARY_DEFINED_BARE_DIE'] * TYPEOF(pd.definition)) > 0) );
    REPEAT k := 1 TO SIZEOF(pd_bag) BY 1;
      IF NOT pass THEN
        ESCAPE;
      END_IF;
      sa := QUERY ( bdt <* USEDIN(pd_bag[k],
```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BARE_DIE_TERMINAL') IN TYPEOF(bdt)) );
    REPEAT l := 1 TO SIZEOF(sa) BY 1;
    IF EXISTS(sa[l]) THEN
        IF sa[l] IN bdt_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            bdt_bag := bdt_bag + sa[l];
        END_IF;
    END_IF;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- part_device_terminal_unique_constraint

RULE part_device_unique_constraint FOR (product_definition_relationship);

LOCAL
    pdr_bag : BAG OF product_definition_relationship;
    name_bag : BAG OF STRING;
    pu_bag : BAG OF physical_unit := [];
    pd : BAG OF product_definition_relationship := QUERY ( pdr < *
        product_definition_relationship | (pdr.description =
        'part device') );
    pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pd) BY 1;
    IF EXISTS(pd[i].relating_product_definition) THEN
        IF (pd[i].relating_product_definition.frame_of_reference.name =
        'physical design usage') AND (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNALLY_DEFINED_PACKAGED_PART',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LIBRARY_DEFINED_PACKAGED_PART'] * TYPEOF(pd[i].
        relating_product_definition)) > 0) THEN
            IF NOT (pd[i].relating_product_definition IN pu_bag) THEN
                pu_bag := pu_bag + pd[i].relating_product_definition;
            END_IF;
        END_IF;
    END_IF;
    END_REPEAT;

```

```

REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  name_bag := [];
  pdr_bag := QUERY ( pdr <* pd | (pdr.relatng_product_definition :=:
    pu_bag[i]) );
  REPEAT j := 1 TO SIZEOF(pdr_bag) BY 1;
    IF EXISTS(pdr_bag[j].name) THEN
      IF pdr_bag[j].name IN name_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        name_bag := name_bag + pdr_bag[j].name;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- part_device_unique_constraint

RULE part_feature_constraint FOR (shape_aspect, product_definition,
  shape_aspect_relationship);

LOCAL
  name_bag : BAG OF STRING := [];
  sa       : BAG OF shape_aspect := [];
  pd       : BAG OF product_definition := QUERY ( pd <*
    product_definition | (pd.frame_of_reference\
    application_context_element.name =
    'physical design usage') );
  pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pd) BY 1;
  name_bag := [];
  sa := QUERY ( sa <* shape_aspect | ((sa.of_shape\property_definition
    .definition = pd[i]) AND sa.product_definitional AND (NOT (sa.
    description = 'connection zone')))) );
  REPEAT j := 1 TO SIZEOF(sa) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    IF sa[j].name IN name_bag THEN
      pass := FALSE;
    ELSE
      name_bag := name_bag + sa[j].name;
    END_IF;
  END_REPEAT;

```

END_REPEAT;

WHERE

WR1: pass;

WR2: (SIZEOF(QUERY (sar <* shape_aspect_relationship | ((sar.name =
'precedent feature') AND (sar.related_shape_aspect = sar.
relating_shape_aspect)))) = 0);

WR3: (SIZEOF(QUERY (sa <* shape_aspect | (((sa.description =
'part group feature') OR (sa.description =
'part generic feature') OR (sa.description =
'polarity indication feature') OR (sa.description =
'interconnect module edge segment surface') OR (sa.description =
'interconnect module edge surface') OR (sa.description =
'interconnect module primary surface') OR (sa.description =
'interconnect module secondary surface') OR (sa.description =
'interconnect module surface feature')) AND (SIZEOF(
QUERY (sar <* USEDIN(sa,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (sar.name
= 'precedent feature'))) > 1)))) = 0);

END_RULE; -- part_feature_constraint

RULE part_feature_unique_constraint FOR (composite_shape_aspect,
shape_aspect, primary_orientation_feature,
secondary_orientation_feature, package_body,
part_tooling_feature, thermal_feature, part_mounting_feature,
package_terminal, assembly_module_terminal, bare_die_terminal,
interconnect_module_terminal,
minimally_defined_bare_die_terminal, packaged_part_terminal,
package_body_surface);

LOCAL

pbs : BAG OF shape_aspect := QUERY (r <* package_body_surface
| TRUE);

ptf : BAG OF shape_aspect := QUERY (r <* part_tooling_feature
| TRUE);

csa : BAG OF shape_aspect := QUERY (r <*
composite_shape_aspect | (r.description =
'part group feature'));

name_bag : BAG OF STRING;

sof : BAG OF shape_aspect := QUERY (r <*
secondary_orientation_feature | TRUE);

tf : BAG OF shape_aspect := QUERY (r <* thermal_feature |
TRUE);

pu_bag : BAG OF physical_unit := [];

pof : BAG OF shape_aspect := QUERY (r <*
primary_orientation_feature | TRUE);

bdt : BAG OF shape_aspect := QUERY (r <* bare_die_terminal |
TRUE);

imt : BAG OF shape_aspect := QUERY (r <*

```

        interconnect_module_terminal | TRUE );
ppt      : BAG OF shape_aspect := QUERY ( r <*
        packaged_part_terminal | TRUE );
amt      : BAG OF shape_aspect := QUERY ( r <*
        assembly_module_terminal | TRUE );
pmf      : BAG OF shape_aspect := QUERY ( r <* part_mounting_feature
        | TRUE );
sa       : BAG OF shape_aspect := QUERY ( r <* shape_aspect | (r.
        description IN ['part generic feature',
        'polarity indication feature',
        'interconnect module edge segment surface',
        'interconnect module edge surface',
        'interconnect module primary surface',
        'interconnect module secondary surface',
        'interconnect module surface feature']) );
pt       : BAG OF shape_aspect := QUERY ( r <* package_terminal |
        TRUE );
mdbdt    : BAG OF shape_aspect := QUERY ( r <*
        minimally_defined_bare_die_terminal | TRUE );
pb       : BAG OF shape_aspect := QUERY ( r <* package_body | TRUE );
pf       : BAG OF shape_aspect := [];
pass     : BOOLEAN := TRUE;
sa_bag   : BAG OF shape_aspect;
END_LOCAL;
sa := csa + sa + pof + sof + pb + ptf + tf + pmf + pt + amt + bdt +
    imt + mdbdt + ppt + pbs;
REPEAT i := 1 TO SIZEOF(sa) BY 1;
    IF EXISTS(sa[i]) THEN
        IF NOT (sa[i] IN pf) THEN
            pf := pf + sa[i];
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pf) BY 1;
    IF EXISTS(pf[i].of_shape.definition) THEN
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT') IN TYPEOF(pf[i].of_shape.definition)) AND (pf[
        i].of_shape.definition.frame_of_reference.name =
        'physical design usage') THEN
            IF NOT (pf[i].of_shape.definition IN pu_bag) THEN
                pu_bag := pu_bag + pf[i].of_shape.definition;
            END_IF;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    name_bag := [];
    sa_bag := QUERY ( sa <* pf | (sa.of_shape.definition :=: pu_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;

```

```

        IF EXISTS(sa_bag[j].name) THEN
            IF sa_bag[j].name IN name_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                name_bag := name_bag + sa_bag[j].name;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- part_feature_unique_constraint

RULE part_template_non_planar_2d_shape_constraint FOR (
    manifold_surface_shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( mssr <* manifold_surface_shape_representation | (
        (mssr.name = 'part template non planar 2d shape') AND (NOT (
            SIZEOF(QUERY ( a2p3d <* mssr.items | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'AXIS2_PLACEMENT_3D') IN TYPEOF(a2p3d)) )) = 1))) )) = 0));

END_RULE; -- part_template_non_planar_2d_shape_constraint

RULE part_template_unique_constraint FOR (part_template_definition);

LOCAL
    name_bag : BAG OF STRING;
    sid_bag  : BAG OF STRING := [];
    pt       : BAG OF part_template_definition := QUERY ( r <*
        part_template_definition | (r.of_shape.definition.
            frame_of_reference.name = 'template definition') );
    pass      : BOOLEAN := TRUE;
    ptd_bag   : BAG OF part_template_definition;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pt) BY 1;
    IF EXISTS(pt[i].source.source_id) THEN
        IF NOT (pt[i].source.source_id IN sid_bag) THEN
            sid_bag := sid_bag + pt[i].source.source_id;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(sid_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;

```



```

name_bag := [];
ptd_bag := QUERY ( ptd <* pt | (ptd.source.source_id = sid_bag[i]) );
REPEAT j := 1 TO SIZEOF(ptd_bag) BY 1;
  IF EXISTS(ptd_bag[j].name) THEN
    IF ptd_bag[j].name IN name_bag THEN
      pass := FALSE;
      ESCAPE;
    ELSE
      name_bag := name_bag + ptd_bag[j].name;
    END_IF;
  END_IF;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- part_template_unique_constraint

RULE part_terminal_external_reference_unique_constraint FOR (
  representation);

LOCAL
  pter      : BAG OF representation := QUERY ( r <* representation | (r.
    name = 'part terminal external reference') );
  pdr_bag   : BAG OF property_definition_representation;
  ft_bag    : BAG OF STRING := [];
  pass      : BOOLEAN := TRUE;
  fn_bag    : BAG OF STRING := [];
  r_bag     : BAG OF representation;
  ptd_bag   : BAG OF part_template_definition;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pter) BY 1;
  REPEAT j := 1 TO SIZEOF(pter[i].items) BY 1;
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(pter[i].items[j])
    THEN
      IF (pter[i].items[j].name = 'feature name') AND (NOT (pter[i].
        items[j].description IN fn_bag)) THEN
        fn_bag := fn_bag + pter[i].items[j].description;
      END_IF;
      IF (pter[i].items[j].name = 'feature type') AND (NOT (pter[i].
        items[j].description IN ft_bag)) THEN
        ft_bag := ft_bag + pter[i].items[j].description;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(fn_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;

```

```

END_IF;
REPEAT j := 1 TO SIZEOF(ft_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  r_bag := QUERY ( r <* pter | ((SIZEOF(QUERY ( ri <* r.items | ((ri
    .description = ft_bag[j]) AND (ri.name = 'feature type')) )) >
    0) AND (SIZEOF(QUERY ( ri <* r.items | ((ri.description =
    fn_bag[i]) AND (ri.name = 'feature name')) )) > 0)) );
  ptd_bag := [];
  REPEAT k := 1 TO SIZEOF(r_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pdr_bag := USEDIN(r_bag[k],
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION');
    REPEAT l := 1 TO SIZEOF(pdr_bag) BY 1;
      IF EXISTS(pdr_bag[l].definition.definition) THEN
        IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
          + 'PHYSICAL_UNIT') IN TYPEOF(pdr_bag[l].definition.
            definition) THEN
          IF pdr_bag[l].definition.definition IN ptd_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            ptd_bag := ptd_bag + pdr_bag[l].definition.definition;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- part_terminal_external_reference_unique_constraint

RULE pdf_requires_security_classification FOR (
  product_definition_formation,
  applied_security_classification_assignment);

WHERE
  WR1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
    SIZEOF(QUERY ( asca <*
    applied_security_classification_assignment | (pdf IN asca.
    items) )) = 1)) )) = 0));

END_RULE; -- pdf_requires_security_classification

```

```

RULE person_and_organization_unique_constraint FOR (
    person_and_organization);

LOCAL
    p_bag    : BAG OF person := [];
    pao_bag  : BAG OF person_and_organization;
    o_bag    : BAG OF organization;
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(person_and_organization) BY 1;
    IF EXISTS(person_and_organization[i].the_person) THEN
        IF NOT (person_and_organization[i].the_person IN p_bag) THEN
            p_bag := p_bag + person_and_organization[i].the_person;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(p_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    o_bag := [];
    pao_bag := QUERY ( pao <* person_and_organization | (pao.the_person
        :=: p_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pao_bag) BY 1;
        IF EXISTS(pao_bag[j].the_organization) THEN
            IF pao_bag[j].the_organization IN o_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                o_bag := o_bag + pao_bag[j].the_organization;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- person_and_organization_unique_constraint

```

```

RULE physical_connectivity_abstraction_map_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    csa_bag : BAG OF component_shape_aspect;
    sar_bag : BAG OF shape_aspect_relationship;
    pcam    : BAG OF shape_aspect_relationship := QUERY ( r <*
        shape_aspect_relationship | (r.name =
            'physical to topological abstraction map') );
    pce_bag : BAG OF physical_connectivity_element := [];

```

```

    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pcam) BY 1;
  IF EXISTS(pcam[i].related_shape_aspect) THEN
    IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'PHYSICAL_CONNECTIVITY_ELEMENT') IN TYPEOF(pcam[i].
        related_shape_aspect) THEN
      IF NOT (pcam[i].related_shape_aspect IN pce_bag) THEN
        pce_bag := pce_bag + pcam[i].related_shape_aspect;
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pce_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  csa_bag := [];
  sar_bag := QUERY ( sar <* pcam | (sar.related_shape_aspect :=:
    pce_bag[i]) );
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'COMPONENT_SHAPE_ASPECT') IN TYPEOF(sar_bag[j].
        relating_shape_aspect)) AND (sar_bag[j].relating_shape_aspect.
        description IN [
          'conductive interconnect element with pre defined transitions',
          'conductive interconnect element with user defined single transition'])
      THEN
        IF EXISTS(sar_bag[j].relating_shape_aspect) THEN
          IF sar_bag[j].relating_shape_aspect IN csa_bag THEN
            pass := FALSE;
            ESCAPE;
          ELSE
            csa_bag := csa_bag + sar_bag[j].relating_shape_aspect;
          END_IF;
        END_IF;
      END_IF;
    END_REPEAT;
  END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- physical_connectivity_abstraction_map_unique_constraint

RULE physical_connectivity_definition_relationship_constraint FOR (
  shape_aspect_relationship);

```

WHERE

```

WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'physical connectivity definition change') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(sar.
    related_shape_aspect)))))) )) = 0);
WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'physical connectivity definition change') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(sar.
    relating_shape_aspect)))))) )) = 0);
WR3: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'physical connectivity definition change') AND (SIZEOF(
    QUERY ( rpd <* USEDIN(sar,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY') IN TYPEOF(rpd)) )) = 0)) )) = 0);

```

END_RULE; -- physical_connectivity_definition_relationship_constraint

RULE physical_connectivity_element_unique_constraint FOR (
 physical_connectivity_element);

LOCAL

```

name_bag : BAG OF STRING := [];
sar_bag : BAG OF shape_aspect_relationship;
pce_bag : BAG OF physical_connectivity_element;
pass : BOOLEAN := TRUE;
pcd_bag : BAG OF physical_connectivity_definition;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(physical_connectivity_element) BY 1;
    IF EXISTS(physical_connectivity_element[i]\shape_aspect.name) THEN
        IF NOT (physical_connectivity_element[i]\shape_aspect.name IN
            name_bag) THEN
            name_bag := name_bag + physical_connectivity_element[i]\
                shape_aspect.name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pcd_bag := [];
    pce_bag := QUERY ( pce <* physical_connectivity_element | (pce\
        shape_aspect.name = name_bag[i]) );
    REPEAT j := 1 TO SIZEOF(pce_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```

sar_bag := QUERY ( sar <* USEDIN(pce_bag[j],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
    = 'structure element') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(sar.
    relating_shape_aspect)))) );
REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].relating_shape_aspect) THEN
        IF sar_bag[j].relating_shape_aspect IN pcd_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            pcd_bag := pcd_bag + sar_bag[j].relating_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

```

WHERE

WR1: pass;

END_RULE; -- physical_connectivity_element_unique_constraint

RULE

physical_node_requirement_to_implementing_component_allocation_constraint
FOR (product_definition_relationship);

WHERE

```

WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
    name = 'physical node requirement to implementation') AND (
    NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION') IN TYPEOF(pdr.
    related_product_definition)))) ) = 0);
WR2: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
    name = 'physical node requirement to implementation') AND (
    SIZEOF(QUERY ( pds <* USEDIN(pdr.relying_product_definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION.DEFINITION') | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(pds)) AND (SIZEOF(
    QUERY ( pcd <* USEDIN(pds,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT.OF_SHAPE') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(pcd)) )) = 0)) ))
    = 0)) )) = 0);

```

END_RULE;

```
-- physical_node_requirement_to_implementing_component_allocation_constraint
```

```
RULE placement_group_area_assignment_constraint FOR (
    shape_aspect_relationship);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'impacted group') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MOUNTING_RESTRICTION_AREA') IN TYPEOF(sar.
    related_shape_aspect)))))) )) = 0);
WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'impacted group') AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_SHAPE_ASPECT') IN TYPEOF(sar.relater_shape_aspect)) OR
    ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT') IN TYPEOF(sar.relater_shape_aspect)) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUP_PRODUCT_DEFINITION') IN TYPEOF(sar.
    relating_shape_aspect.of_shape.definition)) AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_CONTEXT') IN TYPEOF(sar.
    relating_shape_aspect.of_shape.definition.frame_of_reference))
    AND (sar.relater_shape_aspect.of_shape.definition.
    frame_of_reference.name = 'design requirement')))))) )) = 0);
```

```
END_RULE; -- placement_group_area_assignment_constraint
```

```
RULE planar_curve_constraint FOR (shape_representation);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((NOT (sr.name IN [
    'planar projected shape', 'bevel edge feature shape',
    'step edge feature shape', 'radius edge feature shape'])) OR (
    sr\representation.context_of_items\
    geometric_representation_context.coordinate_space_dimension =
    2)) )) = 0);
```

```
END_RULE; -- planar_curve_constraint
```

```
RULE planar_projected_shape_constraint FOR (shape_representation,
    representation_relationship_with_transformation);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT ((NOT (sr\
    representation.name = 'planar projected shape')) OR (SIZEOF(
    QUERY ( rrwt <*
    representation_relationship_with_transformation | (sr :=: rrwt
    \representation_relationship.rep_1) )) <= 1)))) )) = 0);
```

```

END_RULE; -- planar_projected_shape_constraint

RULE planned_effectivity_constraint FOR (configuration_effectivity);

WHERE
  WR1: (SIZEOF(QUERY ( ce <* configuration_effectivity | (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(ce.usage)) AND (ce.usage
    .name = 'assembly composition')))) ) = 0);
  WR2: (SIZEOF(QUERY ( ce <* configuration_effectivity | (SIZEOF(USEDIN(
    ce, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_APPROVAL_ASSIGNMENT.ITEMS')) = 0) )) = 0);

END_RULE; -- planned_effectivity_constraint

RULE planned_effectivity_unique_constraint FOR (
  configuration_effectivity);

LOCAL
  acu_bag : BAG OF assembly_component_usage;
  ci_bag  : BAG OF configuration_item := [];
  ce_bag  : BAG OF configuration_effectivity;
  pass    : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(configuration_effectivity) BY 1;
  IF EXISTS(configuration_effectivity[i].configuration.configuration)
  THEN
    IF NOT (configuration_effectivity[i].configuration.configuration
      IN ci_bag) THEN
      ci_bag := ci_bag + configuration_effectivity[i].configuration.
        configuration;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(ci_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  ce_bag := QUERY ( r <* configuration_effectivity | ((r.configuration
    .configuration :=: ci_bag[i]) AND (r.usage.name =
    'assembly composition') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE') IN TYPEOF(r.usage))) );
  acu_bag := [];
  REPEAT j := 1 TO SIZEOF(ce_bag) BY 1;
    IF EXISTS(ce_bag[j].usage) THEN
      IF ce_bag[j].usage IN acu_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE

```



```

        acu_bag := acu_bag + ce_bag[j].usage;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- planned_effectivity_unique_constraint

RULE plus_minus_tolerance_range_representation_constraint FOR (
    shape_dimension_representation, plus_minus_tolerance);

WHERE
    WR1: (SIZEOF(QUERY ( pmt <* plus_minus_tolerance | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TOLERANCE_VALUE') IN TYPEOF(pmt.range))) )) = 0);

END_RULE; -- plus_minus_tolerance_range_representation_constraint

RULE polarity_indication_feature_constraint FOR (shape_aspect);

WHERE
    WR1: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF(QUERY ( it <*
        QUERY ( sar <* USEDIN(sa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
        = 'associated body vertical extent') AND (sar.
        related_shape_aspect.description =
        'polarity indication feature')) ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY_BOTTOM_SURFACE') IN TYPEOF(it.
        relating_shape_aspect)) )) <= 1)) )) = 0);
    WR2: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF(QUERY ( it <*
        QUERY ( sar <* USEDIN(sa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
        = 'associated body vertical extent') AND (sar.
        related_shape_aspect.description =
        'polarity indication feature')) ) | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PACKAGE_BODY_TOP_SURFACE') IN TYPEOF(it.relateing_shape_aspect)) ))
        <= 1)) )) = 0);
    WR3: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF(QUERY ( it <*
        QUERY ( sar <* USEDIN(sa,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
        = 'associated terminal') AND (sar.related_shape_aspect.
        description = 'polarity indication feature')) ) | ((

```

```
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PACKAGE_TERMINAL') IN TYPEOF(it.relateing_shape_aspect)) )) =
1)) )) = 0);
```

```
END_RULE; -- polarity_indication_feature_constraint
```

```
RULE positional_boundary_member_definition_constraint FOR (
    property_definition);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
    'positional boundary member property') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POSITIONAL_BOUNDARY_MEMBER') IN TYPEOF(pd.definition)))) )) =
    0);
```

```
WR2: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
    'positional boundary member property') AND (NOT (SIZEOF(
    QUERY ( pdr <* USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
    | ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_SIZE_PROPERTY') IN TYPEOF(pdr.
    relateing_property_definition)) AND (pdr.name =
    'boundary member associated dimension')) )) = 1)))) )) = 0);
```

```
END_RULE; -- positional_boundary_member_definition_constraint
```

```
RULE positional_boundary_offset_constraint FOR (property_definition);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
    'positional boundary offset') AND (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP') IN TYPEOF(pd.definition)) AND (pd
    .definition.name =
    'positional boundary and profile boundary member')))) )) = 0);
```

```
WR2: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
    'positional boundary offset') AND (NOT (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATED_PROPERTY_DEFINITION')
    | ((pdr.name = 'boundary offset') AND (pdr.
    relateing_property_definition.name =
    'profile related positional boundary property')) )) = 1)))) ))
    = 0);
```

```
END_RULE; -- positional_boundary_offset_constraint
```

RULE printed_connector_template_terminal_relationship_unique_constraint FOR (
 printed_connector_template_terminal_relationship);

LOCAL

```

  it_bag      : BAG OF printed_part_template_terminal;
  sar_bag      : BAG OF shape_aspect_relationship;
  pass1        : BOOLEAN := TRUE;
  pass2        : BOOLEAN := TRUE;
  pcttr_bag    : BAG OF printed_connector_template_terminal_relationship;
  jt_bag       : BAG OF printed_part_template_terminal;
  ptd_bag      : BAG OF part_template_definition := [];

```

END_LOCAL;

```

REPEAT i := 1 TO SIZEOF(
  printed_connector_template_terminal_relationship) BY 1;
  sar_bag := QUERY ( sar <* USEDIN(
    printed_connector_template_terminal_relationship[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name =
    'connector') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION') IN TYPEOF(sar.relying_shape_aspect))
    AND (sar.relying_shape_aspect.description =
    'printed connector template')) );

```

```

  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].relying_shape_aspect) THEN
      IF NOT (sar_bag[j].relying_shape_aspect IN ptd_bag) THEN
        ptd_bag := ptd_bag + sar_bag[j].relying_shape_aspect;
      END_IF;
    END_IF;

```

END_REPEAT;

END_REPEAT;

```

REPEAT i := 1 TO SIZEOF(ptd_bag) BY 1;

```

```

  IF NOT (pass1 OR pass2) THEN

```

```

    ESCAPE;

```

```

  END_IF;

```

```

  it_bag := [];

```

```

  jt_bag := [];

```

```

  sar_bag := QUERY ( sar <* USEDIN(ptd_bag[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELYING_SHAPE_ASPECT') | (sar.name =
    'connector') );

```

```

  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;

```

```

    IF NOT (pass1 OR pass2) THEN

```

```

      ESCAPE;

```

```

    END_IF;

```

```

    pcttr_bag := QUERY ( pcttr <*
      printed_connector_template_terminal_relationship | (pcttr :=:
      sar_bag[j].related_shape_aspect) );

```

```

    REPEAT k := 1 TO SIZEOF(pcttr_bag) BY 1;

```

```

      IF NOT (pass1 OR pass2) THEN

```

```

        ESCAPE;

```

```

      END_IF;

```

```

IF pass1 AND ((
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(pcttr_bag[k].
  relating_shape_aspect)) AND (pcttr_bag[k].
  relating_shape_aspect.description = 'join terminal') THEN
IF EXISTS(pcttr_bag[k].relating_shape_aspect) THEN
  IF pcttr_bag[k].relating_shape_aspect IN jt_bag THEN
    pass1 := FALSE;
    ESCAPE;
  ELSE
    jt_bag := jt_bag + pcttr_bag[k].relating_shape_aspect;
  END_IF;
END_IF;
END_IF;
IF pass2 AND ((
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(pcttr_bag[k].
  related_shape_aspect)) AND (pcttr_bag[k].related_shape_aspect
  .description = 'interface terminal') THEN
IF EXISTS(pcttr_bag[k].related_shape_aspect) THEN
  IF pcttr_bag[k].related_shape_aspect IN it_bag THEN
    pass2 := FALSE;
    ESCAPE;
  ELSE
    it_bag := it_bag + pcttr_bag[k].related_shape_aspect;
  END_IF;
END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass1;
  WR2: pass2;

END_RULE; -- printed_connector_template_terminal_relationship_unique_constraint

RULE printed_part_template_terminal_relationship_constraint FOR (
  shape_aspect_relationship);

WHERE
  WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'printed part template terminal relationship') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(sar.
    related_shape_aspect)))) ) = 0);
  WR2: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
    'printed part template terminal relationship') AND (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL') IN TYPEOF(sar.

```

```

        relating_shape_aspect)))))) = 0);

END_RULE; -- printed_part_template_terminal_relationship_constraint

RULE printed_part_template_terminal_unique_constraint FOR (
    printed_part_template_terminal);

LOCAL
    pptt      : BAG OF printed_part_template_terminal := QUERY ( r <*
        printed_part_template_terminal | (r.description IN [
            'interface terminal','join terminal']) );
    name_bag   : BAG OF STRING := [];
    sar_bag    : BAG OF shape_aspect_relationship;
    pass       : BOOLEAN := TRUE;
    ptd_bag    : BAG OF part_template_definition;
    sa_bag     : BAG OF shape_aspect;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pptt) BY 1;
    IF EXISTS(pptt[i].name) THEN
        IF NOT (pptt[i].name IN name_bag) THEN
            name_bag := name_bag + pptt[i].name;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(name_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    ptd_bag := [];
    sa_bag := QUERY ( sa <* pptt | (sa.name = name_bag[i]) );
    REPEAT j := 1 TO SIZEOF(sa_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        sar_bag := QUERY ( sar <* USEDIN(sa_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name
            = 'associated definition') AND (sar.relatng_shape_aspect.
            description = 'printed part template')) );
        REPEAT k := 1 TO SIZEOF(sar_bag) BY 1;
            IF EXISTS(sar_bag[k].relating_shape_aspect) THEN
                IF sar_bag[k].relating_shape_aspect IN ptd_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    ptd_bag := ptd_bag + sar_bag[k].relating_shape_aspect;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

```

```

WHERE
    WR1: pass;

END_RULE; -- printed_part_template_terminal_unique_constraint


RULE product_association_constraint FOR (
    product_definition_formation_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdfr <* product_definition_formation_relationship
        | ((pdfr.name = 'product association') AND (SIZEOF(
            QUERY ( adr <* USEDIN(pdfr,
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'EE_SPECIFICATION') IN TYPEOF(adr.assigned_document)) )) = 0)) ))
        = 0));

END_RULE; -- product_association_constraint


RULE product_association_unique_constraint FOR (
    product_definition_formation_relationship);

LOCAL
    es_bag    : BAG OF ee_specification;
    pdfr_bag  : BAG OF product_definition_formation_relationship;
    adr_bag   : BAG OF applied_document_reference;
    pass      : BOOLEAN := TRUE;
    pdfr      : BAG OF product_definition_formation_relationship :=
        QUERY ( r <* product_definition_formation_relationship |
            (r.name = 'product association') );
    pdf_bag   : BAG OF product_definition_formation := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(pdfr) BY 1;
    IF EXISTS(pdfr[i].relating_product_definition_formation) THEN
        IF NOT (pdfr[i].relating_product_definition_formation IN pdf_bag)
            THEN
            pdf_bag := pdf_bag + pdfr[i].
                relating_product_definition_formation;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pdf_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pdfr_bag := QUERY ( r <* pdfr | (r.
        relating_product_definition_formation := pdf_bag[i]) );
    es_bag := [];

```

```

REPEAT j := 1 TO SIZEOF(pdfr_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  adr_bag := QUERY ( adr <* USEDIN(pdfr_bag[j],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_DOCUMENT_REFERENCE.ITEMS') | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EE_SPECIFICATION') IN TYPEOF(adr.assigned_document)) );
  REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
    IF EXISTS(adr_bag[k].assigned_document) THEN
      IF adr_bag[k].assigned_document IN es_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        es_bag := es_bag + adr_bag[k].assigned_document;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- product_association_unique_constraint

RULE product_category_requires_product_related_product_category FOR (
  product_category, product_category_relationship);

LOCAL
  pcr : BAG OF product_category_relationship := [];
  pass : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(product_category) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  pcr := USEDIN(product_category[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY');
  REPEAT j := 1 TO SIZEOF(pcr) BY 1;
    IF NOT pass THEN
      ESCAPE;
    END_IF;
    pass := ((NOT (pcr[j].sub_category.name IN [
      'technology specific land pattern',
      'technology specific padstack',
      'reference packaged part interconnect implementation'])) OR ((
      pcr[j].category.name = 'interconnect module') AND ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PRODUCT_RELATED_PRODUCT_CATEGORY') IN TYPEOF(pcr[j].category))))
AND ((NOT (pcr[j].sub_category.name IN [
    'reference packaged part assembly implementation'])) OR ((pcr[j]
    .category.name = 'assembly module') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_RELATED_PRODUCT_CATEGORY') IN TYPEOF(pcr[j].category))));
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- product_category_requires_product_related_product_category

RULE product_concept_requires_configuration_item FOR (product_concept,
    configuration_item);

WHERE
    WR1: (SIZEOF(QUERY ( pc <* product_concept | (NOT (SIZEOF(
        QUERY ( ci <* configuration_item | (pc :=: ci.item_concept) ))
        >= 1)) )) = 0);

END_RULE; -- product_concept_requires_configuration_item

RULE product_definition_formation_requires_approval FOR (
    product_definition_formation, applied_approval_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
        SIZEOF(QUERY ( aaa <* applied_approval_assignment | (pdf IN
        aaa.items) )) = 1)) )) = 0);

END_RULE; -- product_definition_formation_requires_approval

RULE product_definition_formation_requires_person_organization FOR (
    product_definition_formation,
    applied_person_and_organization_assignment,
    applied_organization_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT ((
        SIZEOF(QUERY ( apaoa <*
        applied_person_and_organization_assignment | ((pdf IN apaoa.
        items) AND (apaoa.role.name = 'creator')) )) = 1) OR (SIZEOF(
        QUERY ( aoa <* applied_organization_assignment | ((pdf IN aoa.
        items) AND (aoa.role.name = 'creator')) )) = 1))) )) = 0);
    WR2: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT ((
        SIZEOF(QUERY ( apaoa <*

```



```

        applied_person_and_organization_assignment | ((pdf IN apaoa.
        items) AND (apaoa.role.name IN ['design supplier',
        'product supplier']))) )) >= 1) OR (SIZEOF(QUERY ( aoa <*
        applied_organization_assignment | ((pdf IN aoa.items) AND (aoa
        .role.name IN ['design supplier','product supplier']))) )) >= 1))) ))
        = 0);

END_RULE; -- product_definition_formation_requires_person_organization

RULE product_definition_requires_date_or_date_and_time FOR (
    product_definition, applied_date_and_time_assignment,
    applied_date_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( pd <* product_definition | (NOT ((SIZEOF(
        QUERY ( adata <* applied_date_and_time_assignment | (pd IN
        adata.items) )) = 1) OR (SIZEOF(QUERY ( ada <*
        applied_date_assignment | (pd IN ada.items) )) = 1)))) )) = 0);

END_RULE; -- product_definition_requires_date_or_date_and_time

RULE product_definition_requires_person_organization FOR (
    product_definition, applied_person_and_organization_assignment,
    applied_organization_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( pd <* product_definition | (NOT ((SIZEOF(
        QUERY ( apaoa <* applied_person_and_organization_assignment |
        (pd IN apaoa.items) )) = 1) OR (SIZEOF(QUERY ( aoa <*
        applied_organization_assignment | (pd IN aoa.items) )) = 1)))) ))
        = 0);

END_RULE; -- product_definition_requires_person_organization

RULE product_requires_person_organization FOR (product,
    applied_person_and_organization_assignment,
    applied_organization_assignment);

WHERE
    WR1: (SIZEOF(QUERY ( prod <* product | (NOT ((SIZEOF(QUERY ( apaoa <*
        applied_person_and_organization_assignment | (prod IN apaoa.
        items) )) = 1) OR (SIZEOF(QUERY ( aoa <*
        applied_organization_assignment | (prod IN aoa.items) )) = 1)))) ))
        = 0);

END_RULE; -- product_requires_person_organization

```

```
RULE product_requires_product_definition_formation FOR (product,
    product_definition_formation);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( prod <* product | (NOT (SIZEOF(QUERY ( pdf <*
        product_definition_formation | (prod :=: pdf.of_product) )) >=
        1)) )) = 0);
```

```
END_RULE; -- product_requires_product_definition_formation
```

```
RULE profile_boundary_definition_with_offsets_constraint FOR (
    property_definition);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
        'boundary zone definition with specified size') AND (pd.name =
        'profile boundary with offsets') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET') IN TYPEOF(pd.
        definition)))))) )) = 0);
```

```
END_RULE; -- profile_boundary_definition_with_offsets_constraint
```

```
RULE profile_related_positional_boundary_definition_constraint FOR (
    property_definition);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
        'profile related positional boundary property') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITIONAL_BOUNDARY') IN TYPEOF(pd.definition)) AND (pd.
        definition.description = 'profile related positional boundary')))) ))
        = 0);
```

```
    WR2: (SIZEOF(QUERY ( pd <* property_definition | ((pd.description =
        'profile related positional boundary property') AND (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
        | ((pdr.related_property_definition.description =
        'positional boundary offset') AND (pdr.name =
        'boundary offset')) )) = 1))) )) = 0);
```

```
END_RULE; -- profile_related_positional_boundary_definition_constraint
```

```
RULE projected_zone_and_base_relationship_constraint FOR (
    projected_zone_definition);
```

```
WHERE
```

```

WR1: (SIZEOF(QUERY ( pzd <* projected_zone_definition | (NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION') IN TYPEOF(pzd.projection_end)) OR
    (pzd.projection_end.product_definitional = TRUE)))) ) = 0);

END_RULE; -- projected_zone_and_base_relationship_constraint

RULE projected_zone_height_characteristic_constraint FOR (
    projected_zone_definition);

WHERE
    WR1: (SIZEOF(QUERY ( pzd <* projected_zone_definition | (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(pzd.projected_length))) )
        = 0);

END_RULE; -- projected_zone_height_characteristic_constraint

RULE promissory_usage_relationship_constraint FOR (
    promissory_usage_occurrence);

WHERE
    WR1: (SIZEOF(QUERY ( puo <* promissory_usage_occurrence | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PHYSICAL_UNIT') IN TYPEOF(puo.relying_product_definition))
        AND (puo.relying_product_definition.frame_of_reference.name =
        'physical design')) ) = 0);
    WR2: (SIZEOF(QUERY ( puo <* promissory_usage_occurrence | (SIZEOF(
        USEDIN(puo,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT.ITEMS')) = 0) )) =
        0);

END_RULE; -- promissory_usage_relationship_constraint

RULE promissory_usage_relationship_unique_constraint FOR (
    promissory_usage_occurrence);

LOCAL
    puo_bag : BAG OF promissory_usage_occurrence;
    pu_bag  : BAG OF physical_unit := [];
    comp_bag : BAG OF physical_unit;
    pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(promissory_usage_occurrence) BY 1;
    IF EXISTS(promissory_usage_occurrence[i].relying_product_definition)
    THEN
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'PHYSICAL_UNIT') IN TYPEOF(promissory_usage_occurrence[i].
        relating_product_definition)) AND (promissory_usage_occurrence[
        i].relating_product_definition.frame_of_reference.name =
        'physical design') THEN
    IF NOT (promissory_usage_occurrence[i].
        relating_product_definition IN pu_bag) THEN
        pu_bag := pu_bag + promissory_usage_occurrence[i].
            relating_product_definition;
    END_IF;
END_IF;
END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(pu_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    puo_bag := QUERY ( puo <* promissory_usage_occurrence | (puo.
        relating_product_definition := pu_bag[i]) );
    comp_bag := [];
    REPEAT j := 1 TO SIZEOF(puo_bag) BY 1;
        IF NOT pass THEN
            ESCAPE;
        END_IF;
        IF (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PHYSICAL_UNIT') IN TYPEOF(puo_bag[j].
            related_product_definition)) AND (puo_bag[j].
            related_product_definition.frame_of_reference.name IN [
            'physical design', 'physical design usage']) THEN
            IF EXISTS(puo_bag[j].related_product_definition) THEN
                IF puo_bag[j].related_product_definition IN comp_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    comp_bag := comp_bag + puo_bag[j].related_product_definition;
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- promissory_usage_relationship_unique_constraint

RULE radius_edge_feature_shape_constraint FOR (shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'radius edge feature shape') AND (SIZEOF(QUERY ( lmwu <* sr.
        items | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'maximum radius length')) )) = 0)) )) = 0);
WR2: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'radius edge feature shape') AND (SIZEOF(QUERY ( lmwu <* sr.
        items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'minimum radius length')) )) = 0)) )) = 0);

END_RULE; -- radius_edge_feature_shape_constraint

RULE reference_composition_path_constraint FOR (
    product_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name = 'reference composition path') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_FUNCTIONAL_UNIT') IN TYPEOF(pdr.
        relating_product_definition)) AND (pdr.
        relating_product_definition.description =
        'reference definition path') AND (pdr.
        relating_product_definition.frame_of_reference.name =
        'functional occurrence')))) )) = 0);
    WR2: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name = 'reference composition path') AND (NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_FUNCTIONAL_UNIT') IN TYPEOF(pdr.
        related_product_definition)) AND (pdr.
        related_product_definition.frame_of_reference.name =
        'functional occurrence')))) )) = 0);

END_RULE; -- reference_composition_path_constraint

RULE reference_functional_unit_assignment_to_part_constraint FOR (
    product_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
        name = 'reference functional unit assignment to part') AND (
        NOT (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION') IN TYPEOF(pdr.related_product_definition))
        AND (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'APPLICATION_CONTEXT_ELEMENT') IN TYPEOF(pdr.
        related_product_definition.frame_of_reference)) AND ((pdr.
        relating_product_definition.frame_of_reference.name =
        'physical design') OR (pdr.relatng_product_definition.
        frame_of_reference.name = 'physical design usage')))) )) = 0);

```

```

WR2: (SIZEOF(QUERY ( pdr <* product_definition_relationship | ((pdr.
    name = 'reference functional unit assignment to part') AND (
    NOT (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION') IN TYPEOF(pdr.
    relating_product_definition)) AND (pdr.
    relating_product_definition.description =
    'design definition path') AND (pdr.relatng_product_definition
    .id = 'reference composition path')))) )) = 0);

```

```

END_RULE; -- reference_functional_unit_assignment_to_part_constraint

```

```

RULE reference_functional_unit_assignment_to_part_unique_constraint FOR (
    shape_aspect_relationship);

```

```

LOCAL
    sar      : BAG OF shape_aspect_relationship := QUERY ( r <*
        shape_aspect_relationship | (r.name =
        'reference functional unit assignment to part') );
    pd_bag   : BAG OF product_definition;
    sar_bag   : BAG OF shape_aspect_relationship;
    pass      : BOOLEAN := TRUE;
    d_bag     : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(sar) BY 1;
    IF EXISTS(sar[i].description) THEN
        IF NOT (sar[i].description IN d_bag) THEN
            d_bag := d_bag + sar[i].description;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(d_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sar_bag := QUERY ( r <* sar | (r.description = d_bag[i]) );
    pd_bag := [];
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect.of_shape.definition)
            THEN
                IF sar_bag[j].related_shape_aspect.of_shape.definition IN pd_bag
                    THEN
                        pass := FALSE;
                        ESCAPE;
                    ELSE
                        pd_bag := pd_bag + sar_bag[j].related_shape_aspect.of_shape.
                            definition;
                    END_IF;
                END_IF;
            END_REPEAT;
        END_REPEAT;
    END_REPEAT;

```

```

WHERE
    WR1: pass;

END_RULE; -- reference_functional_unit_assignment_to_part_unique_constraint

RULE registered_font_constraint FOR (externally_defined_text_font);

WHERE
    WR1: (SIZEOF(QUERY ( edtf <* externally_defined_text_font | (SIZEOF(
        QUERY ( adr <* USEDIN(edtf,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (adr.assigned_document.
            kind.product_data_type = 'font registration document') )) = 0) ))
        = 0);

END_RULE; -- registered_font_constraint

RULE registered_font_unique_constraint FOR (externally_defined_text_font);

LOCAL
    ii_bag    : BAG OF STRING := [];
    es_bag    : BAG OF ee_specification;
    edtf_bag  : BAG OF externally_defined_text_font;
    adr_bag   : BAG OF applied_document_reference;
    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(externally_defined_text_font) BY 1;
    IF EXISTS(externally_defined_text_font[i].item_id) THEN
        IF NOT (externally_defined_text_font[i].item_id IN ii_bag) THEN
            ii_bag := ii_bag + externally_defined_text_font[i].item_id;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(ii_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    edtf_bag := QUERY ( edtf <* externally_defined_text_font | (edtf.
        item_id = ii_bag[i]) );
    REPEAT j := 1 TO SIZEOF(edtf_bag) BY 1;
        adr_bag := QUERY ( adr <* USEDIN(edtf_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS') | (adr.assigned_document.
            kind.product_data_type = 'font registration document') );
    es_bag := [];
    REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
        IF EXISTS(adr_bag[k].assigned_document) THEN
            IF adr_bag[k].assigned_document IN es_bag THEN
                pass := FALSE;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```

```

        ESCAPE;
    ELSE
        es_bag := es_bag + adr_bag[k].assigned_document;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- registered_font_unique_constraint


RULE registered_style_constraint FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'registered case style') AND (SIZEOF(QUERY ( dri <* rep.items
        | ((( 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(dri)) AND (dri.
        name = 'case style code')) )) = 0)) )) = 0);
    WR2: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
        'registered case style') AND (SIZEOF(USEDIN(rep,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 0)) )) = 0);

END_RULE; -- registered_style_constraint


RULE registered_style_unique_constraint FOR (representation);

LOCAL
    es_bag : BAG OF ee_specification;
    rs      : BAG OF representation := QUERY ( r <* representation | (r.
        name = 'registered case style') );
    adr_bag : BAG OF applied_document_reference;
    dri_bag : BAG OF descriptive_representation_item := [];
    pass    : BOOLEAN := TRUE;
    r_bag   : BAG OF representation;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(rs) BY 1;
    REPEAT j := 1 TO SIZEOF(rs[i].items) BY 1;
        IF (rs[i].items[j].name = 'case style code') AND ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(rs[i].items[j]))
        THEN
            IF EXISTS(rs[i].items[j]) THEN
                IF NOT (rs[i].items[j] IN dri_bag) THEN
                    dri_bag := dri_bag + rs[i].items[j];
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

```



```

        END_IF;
    END_IF;
END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(dri_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    r_bag := QUERY ( r <* rs | (dri_bag[i] IN r.items) );
    REPEAT j := 1 TO SIZEOF(r_bag) BY 1;
        adr_bag := USEDIN(r_bag[j],
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'APPLIED_DOCUMENT_REFERENCE.ITEMS');
        es_bag := [];
        REPEAT k := 1 TO SIZEOF(adr_bag) BY 1;
            IF EXISTS(adr_bag[k].assigned_document) THEN
                IF adr_bag[k].assigned_document IN es_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    es_bag := es_bag + adr_bag[k].assigned_document;
                END_IF;
            END_IF;
        END_REPEAT;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- registered_style_unique_constraint

RULE requirement_composition_constraint FOR (
    property_definition_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* property_definition_relationship | ((pdr.
        name = 'requirements property composition') AND (NOT ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
            related_property_definition)))) ) = 0);
    WR2: (SIZEOF(QUERY ( pdr <* property_definition_relationship | ((pdr.
        name = 'requirements property composition') AND (NOT ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REQUIREMENTS_PROPERTY_GROUP') IN TYPEOF(pdr.
            relating_property_definition)))) ) = 0);

END_RULE; -- requirement_composition_constraint

RULE restraint_condition_constraint FOR (property_definition);

```

```

WHERE
  WR1: (SIZEOF(QUERY ( pd <* property_definition | (((pd.description =
    'restraint') OR (pd.description =
    'tolerance specific restraint')) AND (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION')
    | (((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
    + 'REQUIREMENTS_PROPERTY') IN TYPEOF(pdr.
    related_property_definition)) AND (pdr.name =
    'restraint description')) )) = 0)) )) = 0);

```

```
END_RULE; -- restraint_condition_constraint
```

```
RULE restrict_action_request_status FOR (action_request_status);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( ars <* action_request_status | (NOT (ars.status
    IN ['proposed', 'in work', 'issued', 'hold']))) )) = 0);

```

```
END_RULE; -- restrict_action_request_status
```

```
RULE restrict_approval_status FOR (approval_status);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( ast <* approval_status | (NOT (ast.name IN [
    'approved', 'not yet approved', 'disapproved', 'withdrawn']))) ))
    = 0);

```

```
END_RULE; -- restrict_approval_status
```

```
RULE restrict_date_role FOR (date_role);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( dr <* date_role | (NOT (dr.name IN [
    'creation date', 'request date', 'release date', 'start date',
    'contract date', 'certification date', 'sign off date',
    'classification date', 'declassification date']))) )) = 0);

```

```
END_RULE; -- restrict_date_role
```

```
RULE restrict_manifold_surface_shape_representation FOR (
  manifold_surface_shape_representation);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( mssr <* manifold_surface_shape_representation | (

```

```

(mssr.name IN ['stratum feature non planar 2d shape',
'open shell based surface',
'design intent modification non planar 2d shape',
'part template non planar 2d shape']) AND (NOT (SIZEOF(
QUERY ( sbsm <* mssr.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHELL_BASED_SURFACE_MODEL') IN TYPEOF(sbsm)) AND (SIZEOF(
QUERY ( os <* sbsm\shell_based_surface_model.sbsm_boundary | (
('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'OPEN_SHELL') IN TYPEOF(os)) )) = 1)) )) = 1))) )) = 0);

```

```
END_RULE; -- restrict_manifold_surface_shape_representation
```

```
RULE rule_action_unique_constraint FOR (rule_action);
```

```
LOCAL
```

```

rd_bag : BAG OF rule_definition;
aaa_bag : BAG OF applied_action_assignment;
adta : BAG OF applied_date_and_time_assignment;
pass : BOOLEAN := TRUE;
ra_bag : BAG OF rule_action;
adta_bag : BAG OF applied_date_and_time_assignment := [];

```

```
END_LOCAL;
```

```
REPEAT i := 1 TO SIZEOF(rule_action) BY 1;
```

```

adta := USEDIN(rule_action[i],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS');

```

```
REPEAT j := 1 TO SIZEOF(adta) BY 1;
```

```

IF EXISTS(adta[j]) THEN
  IF NOT (adta[j] IN adta_bag) THEN
    adta_bag := adta_bag + adta[j];
  END_IF;
END_IF;

```

```

END_REPEAT;
END_REPEAT;

```

```

REPEAT i := 1 TO SIZEOF(adta_bag) BY 1;
IF NOT pass THEN

```

```

  ESCAPE;
END_IF;

```

```

ra_bag := QUERY ( r <* rule_action | (r IN adta_bag[i].items) );
rd_bag := [];

```

```

REPEAT j := 1 TO SIZEOF(ra_bag) BY 1;
IF NOT pass THEN

```

```

  ESCAPE;
END_IF;

```

```

aaa_bag := QUERY ( aa <* USEDIN(ra_bag[j],
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'APPLIED_ACTION_ASSIGNMENT') IN TYPEOF(aa)) );

```

```
REPEAT k := 1 TO SIZEOF(aaa_bag) BY 1;
```

```

    IF NOT pass THEN
        ESCAPE;
    END_IF;
    REPEAT l := 1 TO SIZEOF(aaa_bag[k].items) BY 1;
        IF EXISTS(aaa_bag[k].items[l]) THEN
            IF ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
                + 'RULE_DEFINITION') IN TYPEOF(aaa_bag[k].items[l]) THEN
                IF aaa_bag[k].items[l] IN rd_bag THEN
                    pass := FALSE;
                    ESCAPE;
                ELSE
                    rd_bag := rd_bag + aaa_bag[k].items[l];
                END_IF;
            END_IF;
        END_IF;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;
    END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- rule_action_unique_constraint


RULE rule_priority_constraint FOR (representation);

LOCAL
    rpc_group_1 : SET OF representation := QUERY ( rule_group_1 <*
        representation | (rule_group_1.name = 'rule property') );
END_LOCAL;

WHERE
    WR1: (SIZEOF(QUERY ( x <* rpc_group_1 | (NOT (SIZEOF(QUERY ( y <* x.
        items | ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(y)) )) = 1)) )) =
        0);
    WR2: (SIZEOF(QUERY ( x <* rpc_group_1 | (NOT (SIZEOF(QUERY ( y <*
        USEDIN(x,
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.' + 'USED_REPRESENTATION')
            | ((('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
            + 'RULE_DEFINITION') IN TYPEOF(y.definition.definition)) )) >=
            1)) )) = 0);

END_RULE; -- rule_priority_constraint


RULE security_classification_constraint FOR (security_classification,
    applied_date_and_time_assignment, applied_date_assignment,

```

```

        applied_person_assignment, applied_approval_assignment);

WHERE
  WR1: (SIZEOF(QUERY ( sc <* security_classification | (NOT ((SIZEOF(
    QUERY ( adata <* applied_date_and_time_assignment | ((sc IN
    adata.items) AND ('classification date' = adata.role.name)) ))
    = 1) OR (SIZEOF(QUERY ( ada <* applied_date_assignment | ((sc
    IN ada.items) AND ('classification date' = ada.role.name)) ))
    = 1))) )) = 0);
  WR2: (SIZEOF(QUERY ( sc <* security_classification | (SIZEOF(
    QUERY ( apa <* applied_person_assignment | ((sc IN apa.items)
    AND (apa.role.name = 'classification officer')) )) = 0) )) = 0);
  WR3: (SIZEOF(QUERY ( sc <* security_classification | (SIZEOF(
    QUERY ( aaa <* applied_approval_assignment | (sc IN aaa.items) ))
    = 0) )) = 0);

END_RULE; -- security_classification_constraint

```

```

RULE shape_dimension_representation_constraint FOR (
  shape_dimension_representation);

```

```

WHERE
  WR1: (SIZEOF(QUERY ( sdr <* shape_dimension_representation | (NOT (
    SIZEOF(QUERY ( i <* sdr.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_WITH_UNIT') IN TYPEOF(i)) AND (SIZEOF(QUERY ( mq <*
    USEDIN(i,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_QUALIFICATION.QUALIFIED_MEASURE') | (mq.name =
    'dimension value qualifier') )) <= 1)) )) >= 1)) )) = 0);
  WR2: (SIZEOF(QUERY ( sdr <* shape_dimension_representation | (SIZEOF(
    QUERY ( i <* sdr.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_WITH_UNIT') IN TYPEOF(i)) AND (SIZEOF(QUERY ( mq <*
    USEDIN(i,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_QUALIFICATION.QUALIFIED_MEASURE') | ((mq.name =
    'dimension value qualifier') AND (SIZEOF(QUERY ( q <* mq.
    qualifiers | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TYPE_QUALIFIER') IN TYPEOF(q)) )) <> 1)) )) > 0)) )) > 0) ))
    = 0);
  WR3: (SIZEOF(QUERY ( sdr <* shape_dimension_representation | (SIZEOF(
    QUERY ( i <* sdr.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_WITH_UNIT') IN TYPEOF(i)) AND (SIZEOF(QUERY ( mq <*
    USEDIN(i,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MEASURE_QUALIFICATION.QUALIFIED_MEASURE') | ((mq.name =
    'dimension value qualifier') AND (SIZEOF(QUERY ( q <* mq.
    qualifiers | (((

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TYPE_QUALIFIER') IN TYPEOF(q)) AND (NOT (q.name IN [
        'theoretically exact', 'maximum dimension', 'minumum dimension',
        'lower value', 'upper value', 'basic value']))) )) > 0)) )) > 0)) ))
        > 0)) = 0);
WR4: (SIZEOF(QUERY ( sdr <* shape_dimension_representation | (SIZEOF(
        QUERY ( i <* sdr.items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_WITH_UNIT') IN TYPEOF(i)) AND (SIZEOF(QUERY ( mq <*
        USEDIN(i,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'MEASURE_QUALIFICATION.QUALIFIED_MEASURE') | (NOT (mq.name IN
        ['dimension value qualifier', 'predefined dimension qualifier',
        'user defined dimension qualifier']))) )) > 0)) )) > 0)) = 0);
WR5: (SIZEOF(QUERY ( sdr <* shape_dimension_representation | (NOT (
        SIZEOF(USEDIN(sdr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DIMENSIONAL_CHARACTERISTIC_REPRESENTATION.' +
        'REPRESENTATION')) >= 1)) )) = 0);

END_RULE; -- shape_dimension_representation_constraint

RULE shape_element_composing_relationship_constraint FOR (
    shape_aspect_relationship);

WHERE
    WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | ((sar.name =
        'composing') AND ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_SHAPE_ASPECT') IN TYPEOF(sar.relateing_shape_aspect))) ))
        = 0);

END_RULE; -- shape_element_composing_relationship_constraint

RULE shape_element_constituent_relationship_unique_constraint FOR (
    shape_aspect_relationship);

LOCAL
    sar      : BAG OF shape_aspect_relationship := QUERY ( r <*
        shape_aspect_relationship | (r.name IN ['constituent',
        'composing']) );
    sar_bag  : BAG OF shape_aspect_relationship;
    pass     : BOOLEAN := TRUE;
    rsa_bag  : BAG OF shape_aspect;
    sa_bag   : BAG OF shape_aspect := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(sar) BY 1;
    IF EXISTS(sar[i].relateing_shape_aspect) THEN
        IF NOT (sar[i].relateing_shape_aspect IN sa_bag) THEN
            sa_bag := sa_bag + sar[i].relateing_shape_aspect;

```

```

        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(sa_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sar_bag := QUERY ( r <* sar | (r.relatng_shape_aspect :=: sa_bag[i]) );
    rsa_bag := [];
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
        IF EXISTS(sar_bag[j].related_shape_aspect) THEN
            IF sar_bag[j].related_shape_aspect IN rsa_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                rsa_bag := rsa_bag + sar_bag[j].related_shape_aspect;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- shape_element_constituent_relationship_unique_constraint

```

RULE shape_element_constraint FOR (shape_aspect);

```

LOCAL
    name_bag : BAG OF STRING := [];
    sab      : BAG OF shape_aspect := QUERY ( sa <* shape_aspect | ((sa.
        name IN [
'conductive interconnect element with user defined single transition',
'constrained intra layer join','intra stratum join',
'inter stratum join','common datum','single datum',''])
OR (SIZEOF(TYPEOF(sa)) >= 1) OR (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(sa.of_shape.
definition)) AND (sa.of_shape.definition.
frame_of_reference.name = 'physical design usage')))) );
    pass2    : BOOLEAN := TRUE;
    sa       : BAG OF shape_aspect := QUERY ( sa <* shape_aspect | ((
        NOT (sa.name IN [
'conductive interconnect element with user defined single transition',
'constrained intra layer join','intra stratum join',
'inter stratum join','common datum','single datum',''])
OR (SIZEOF(TYPEOF(sa)) = 1) OR (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
+ 'PRODUCT_DEFINITION_SHAPE') IN TYPEOF(sa.of_shape.
definition)) AND (sa.of_shape.definition.
frame_of_reference.name = 'physical design usage')))) ) );

```

```

    pass      : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(sa) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF sa[i].name IN name_bag THEN
        pass := FALSE;
    ELSE
        name_bag := name_bag + sa[i].name;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(sab) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF NOT pass2 THEN
        ESCAPE;
    END_IF;
    IF sab[i].name IN name_bag THEN
        pass2 := FALSE;
    ELSE
        name_bag := name_bag + sab[i].name;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;
    WR2: pass2;

END_RULE; -- shape_element_constraint

RULE shape_element_deriving_relationship_unique_constraint FOR (
    shape_aspect_deriving_relationship);

LOCAL
    sadr_bag : BAG OF shape_aspect_deriving_relationship;
    pass     : BOOLEAN := TRUE;
    rsa_bag  : BAG OF shape_aspect;
    sa_bag   : BAG OF shape_aspect := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(shape_aspect_deriving_relationship) BY 1;
    IF EXISTS(shape_aspect_deriving_relationship[i].
        relating_shape_aspect) THEN
        IF NOT (shape_aspect_deriving_relationship[i].
            relating_shape_aspect IN sa_bag) THEN
            sa_bag := sa_bag + shape_aspect_deriving_relationship[i].
                relating_shape_aspect;
        END_IF;
    END_IF;
END_REPEAT;

```



```

REPEAT i := 1 TO SIZEOF(sa_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  sadr_bag := QUERY ( r <* shape_aspect_deriving_relationship | (r.
    relating_shape_aspect :=: sa_bag[i]) );
  rsa_bag := [];
  REPEAT j := 1 TO SIZEOF(sadr_bag) BY 1;
    IF EXISTS(sadr_bag[j].related_shape_aspect) THEN
      IF sadr_bag[j].related_shape_aspect IN rsa_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        rsa_bag := rsa_bag + sadr_bag[j].related_shape_aspect;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- shape_element_deriving_relationship_unique_constraint

RULE shape_element_locating_relationship_unique_constraint FOR (
  dimensional_location);

LOCAL
  dl_bag : BAG OF dimensional_location;
  pass : BOOLEAN := TRUE;
  rsa_bag : BAG OF shape_aspect;
  sa_bag : BAG OF shape_aspect := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(dimensional_location) BY 1;
  IF EXISTS(dimensional_location[i].relating_shape_aspect) THEN
    IF NOT (dimensional_location[i].relating_shape_aspect IN sa_bag)
      THEN
      sa_bag := sa_bag + dimensional_location[i].relating_shape_aspect;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(sa_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  dl_bag := QUERY ( r <* dimensional_location | (r.
    relating_shape_aspect :=: sa_bag[i]) );
  rsa_bag := [];
  REPEAT j := 1 TO SIZEOF(dl_bag) BY 1;
    IF EXISTS(dl_bag[j].related_shape_aspect) THEN
      IF dl_bag[j].related_shape_aspect IN rsa_bag THEN

```

```

        pass := FALSE;
        ESCAPE;
    ELSE
        rsa_bag := rsa_bag + dl_bag[j].related_shape_aspect;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- shape_element_locating_relationship_unique_constraint

RULE shape_element_unique_constraint FOR (representation,
    representation_item, property_definition_representation);

LOCAL
    r          : BAG OF representation := QUERY ( r <*
        representation | (r\representation.name =
            'shape element name') );
    description_bag : BAG OF STRING := [];
    ri          : BAG OF representation_item := [];
    pass        : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pass := SIZEOF(QUERY ( pdr <* USEDIN(r[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | (pdr.
        definition.description = 'shape element characterization') )) = 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    ri := QUERY ( ri <* representation_item | (ri IN r[i].items) );
    pass := SIZEOF(ri) = 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    IF NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM') IN TYPEOF(ri)) THEN
        pass := FALSE;
        ESCAPE;
    END_IF;
    REPEAT j := 1 TO SIZEOF(ri) BY 1;
        IF EXISTS(ri[j].description) THEN
            IF ri[j]\representation_item.description IN description_bag
                THEN
                pass := FALSE;
            END_IF;
        END_REPEAT;
    END_REPEAT;

```

```

        ESCAPE;
    ELSE
        description_bag := description_bag + ri[j].description;
    END_IF;
END_IF;
END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- shape_element_unique_constraint

RULE shape_representation_requires_origin FOR (shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT ((SIZEOF(
        QUERY ( it <* sr\representation.items | (((it.name = 'origin')
        OR (it.name = 'orientation')) AND (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D') IN TYPEOF(it)) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D') IN TYPEOF(it)))) )) = 1) OR (SIZEOF(
        QUERY ( it <* sr\representation.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STYLED_ITEM') IN TYPEOF(it)) )) >= 1) OR (SIZEOF(
        QUERY ( it <* sr\representation.items | (((it.name = 'origin')
        OR (it.name = 'orientation')) AND ((sr.name = 'zone shape') OR
        (sr.context_of_items.context_type = 'connection zone colour')))) ))
        = 0))) )) = 0);

END_RULE; -- shape_representation_requires_origin

RULE step_edge_feature_shape_constraint FOR (shape_representation);

WHERE
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'step edge feature shape') AND (NOT (SIZEOF(QUERY ( lmwu <* sr
        .items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'maximum step depth')) )) = 1)))) )) = 0);
    WR2: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'step edge feature shape') AND (NOT (SIZEOF(QUERY ( lmwu <* sr
        .items | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
        'maximum step width')) )) = 1)))) )) = 0);
    WR3: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
        'step edge feature shape') AND (NOT (SIZEOF(QUERY ( lmwu <* sr

```

```

.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
'minimum step depth')) )) = 1))) )) = 0);
WR4: (SIZEOF(QUERY ( sr <* shape_representation | ((sr.name =
'step edge feature shape') AND (NOT (SIZEOF(QUERY ( lmwu <* sr
.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
'minimum step width')) )) = 1))) )) = 0);

END_RULE; -- step_edge_feature_shape_constraint

```

```

RULE stratum_average_surface_shape_constraint FOR (
    manifold_surface_shape_representation);

```

```

WHERE
    WR1: (SIZEOF(QUERY ( mssr <* manifold_surface_shape_representation | (
        (mssr.name = 'stratum average surface shape') AND (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(mssr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        ('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_SURFACE') IN TYPEOF(pdr.definition.definition)) AND (
        pdr.definition.definition.description = 'average surface')) ))
        = 1))) )) = 0);

```

```

END_RULE; -- stratum_average_surface_shape_constraint

```

```

RULE stratum_feature_non_planar_2d_shape_constraint FOR (
    manifold_surface_shape_representation);

```

```

WHERE
    WR1: (SIZEOF(QUERY ( mssr <* manifold_surface_shape_representation | (
        (mssr.name = 'stratum feature non planar 2d shape') AND (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(mssr,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_FEATURE') IN TYPEOF(pdr.definition.definition)) ))
        = 1))) )) = 0);

```

```

END_RULE; -- stratum_feature_non_planar_2d_shape_constraint

```

```

RULE
stratum_feature_shape_stratum_average_surface_shape_relationship_constraint
FOR ( representation_relationship);

```

WHERE

```

WR1: (SIZEOF(QUERY ( rr <* representation_relationship | ((rr.name =
'stratum feature shape stratum average surface shape relationship')
AND (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MANIFOLD_SURFACE_SHAPE_REPRESENTATION') IN TYPEOF(rr.rep_2))
AND (rr.rep_2.name = 'stratum feature non planar 2d shape')))) ))
= 0);
WR2: (SIZEOF(QUERY ( rr <* representation_relationship | ((rr.name =
'stratum feature shape stratum average surface shape relationship')
AND (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MANIFOLD_SURFACE_SHAPE_REPRESENTATION') IN TYPEOF(rr.rep_1))
AND (rr.rep_1.name = 'stratum average surface shape')))) )) =
0);

```

END_RULE;

-- stratum_feature_shape_stratum_average_surface_shape_relationship_constraint

```

RULE stratum_feature_shape_stratum_surface_shape_relationship_constraint
FOR ( representation_relationship);

```

WHERE

```

WR1: (SIZEOF(QUERY ( rr <* representation_relationship | ((rr.name =
'stratum feature shape stratum surface shape relationship')
AND (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MANIFOLD_SURFACE_SHAPE_REPRESENTATION') IN TYPEOF(rr.rep_2))
AND (rr.rep_2.name = 'stratum feature non planar 2d shape')))) ))
= 0);
WR2: (SIZEOF(QUERY ( rr <* representation_relationship | ((rr.name =
'stratum feature shape stratum surface shape relationship')
AND (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MANIFOLD_SURFACE_SHAPE_REPRESENTATION') IN TYPEOF(rr.rep_1))
AND (rr.rep_1.name = 'stratum surface shape')))) )) = 0);

```

END_RULE;

-- stratum_feature_shape_stratum_surface_shape_relationship_constraint

```

RULE stratum_feature_unique_constraint FOR (stratum_feature);

```

LOCAL

```

name_bag : BAG OF STRING;
s_bag    : BAG OF stratum := [];
sf_bag   : BAG OF stratum_feature;
pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(stratum_feature) BY 1;
  IF EXISTS(stratum_feature[i].of_shape.definition) THEN

```

```

        IF NOT (stratum_feature[i].of_shape.definition IN s_bag) THEN
            s_bag := s_bag + stratum_feature[i].of_shape.definition;
        END_IF;
    END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(s_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    sf_bag := QUERY ( r <* stratum_feature | (r.of_shape.definition :=:
        s_bag[i]) );
    name_bag := [];
    REPEAT j := 1 TO SIZEOF(sf_bag) BY 1;
        IF EXISTS(sf_bag[j].name) THEN
            IF sf_bag[j].name IN name_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                name_bag := name_bag + sf_bag[j].name;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- stratum_feature_unique_constraint

RULE stratum_interconnect_module_3d_position_constraint FOR (mapped_item);

WHERE
    WR1: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
        '3d positioned stratum shape in interconnect module') AND (
        NOT (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.'
        + 'AXIS2_PLACEMENT_3D') IN TYPEOF(mi.mapping_target)))) ) = 0);

END_RULE; -- stratum_interconnect_module_3d_position_constraint

RULE stratum_interconnect_module_planar_position_constraint FOR (
    mapped_item);

WHERE
    WR1: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
        'planar positioned stratum shape in interconnect module') AND
        (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D') IN TYPEOF(mi.mapping_target)))) ) = 0);

END_RULE; -- stratum_interconnect_module_planar_position_constraint

```

```
RULE stratum_make_from_relationship_constraint FOR (
    make_from_usage_option);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( mfuo <* make_from_usage_option | ((mfuo.name =
        'stratum make from') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM') IN TYPEOF(mfuo.related_product_definition)))) )) =
        0);
    WR2: (SIZEOF(QUERY ( mfuo <* make_from_usage_option | ((mfuo.name =
        'stratum make from') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM') IN TYPEOF(mfuo.relatng_product_definition)))) )) =
        0);
```

```
END_RULE; -- stratum_make_from_relationship_constraint
```

```
RULE stratum_planar_shape_constraint FOR (shape_representation);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT ((NOT ((sr.name
        = 'planar projected shape') AND (SIZEOF(QUERY ( pdr <* USEDIN(
        sr, 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM') IN TYPEOF(pdr.definition.definition)) )) = 1))) OR
        (SIZEOF(QUERY ( a2p2d <* sr.items | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_2D') IN TYPEOF(a2p2d)) )) >= 0))) )) = 0);
```

```
END_RULE; -- stratum_planar_shape_constraint
```

```
RULE stratum_surface_in_stratum_3d_position_constraint FOR (mapped_item);
```

```
WHERE
```

```
    WR1: (SIZEOF(QUERY ( mi <* mapped_item | ((mi.name =
        'stratum surface in stratum 3d position') AND (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'AXIS2_PLACEMENT_3D') IN TYPEOF(mi.mapping_target)))) )) = 0);
```

```
END_RULE; -- stratum_surface_in_stratum_3d_position_constraint
```

```
RULE stratum_surface_unique_constraint FOR (stratum_surface);
```

```
    LOCAL
```

```
        desc_bag : BAG OF STRING;
        s_bag     : BAG OF stratum := [];
```

```

    pass      : BOOLEAN := TRUE;
    ss_bag    : BAG OF stratum_surface;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(stratum_surface) BY 1;
  IF EXISTS(stratum_surface[i].of_shape.definition) THEN
    IF NOT (stratum_surface[i].of_shape.definition IN s_bag) THEN
      s_bag := s_bag + stratum_surface[i].of_shape.definition;
    END_IF;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(s_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  ss_bag := QUERY ( r <* stratum_surface | (r.of_shape.definition ::=
    s_bag[i]) );
  desc_bag := [];
  REPEAT j := 1 TO SIZEOF(ss_bag) BY 1;
    IF EXISTS(ss_bag[j].description) THEN
      IF ss_bag[j].description IN desc_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        desc_bag := desc_bag + ss_bag[j].description;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- stratum_surface_unique_constraint

RULE stratum_technology_line_width_tolerance_assignment_constraint FOR (
  representation);

WHERE
  WR1: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
    'stratum technology line width tolerance assignment') AND (
    NOT (SIZEOF(QUERY ( lmwu <* rep.items | (((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
    'maximum positive deviation')) )) = 1))) )) = 0);
  WR2: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
    'stratum technology line width tolerance assignment') AND (
    NOT (SIZEOF(QUERY ( si <* rep.items | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STYLED_ITEM') IN TYPEOF(si)) )) = 1))) )) = 0);
  WR3: (SIZEOF(QUERY ( rep <* representation | ((rep.name =
    'stratum technology line width tolerance assignment') AND (

```



```

NOT (SIZEOF(QUERY ( lmwu <* rep.items | (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LENGTH_MEASURE_WITH_UNIT') IN TYPEOF(lmwu)) AND (lmwu.name =
'maximum negative deviation')) )) = 1))) )) = 0);

END_RULE; -- stratum_technology_line_width_tolerance_assignment_constraint

RULE substitute_product_constraint FOR (product,
assembly_component_usage_substitute);

WHERE
WR1: (SIZEOF(QUERY ( acus <* assembly_component_usage_substitute |
EXISTS(acus.substitute.reference_designator) )) = 0);

END_RULE; -- substitute_product_constraint

RULE subtype_combination_shape_aspect FOR (shape_aspect);

WHERE
WR1: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_BOND_DEFINITION',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MINIMALLY_DEFINED_BARE_DIE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_FUNCTIONAL_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_INTERFACE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONNECTIVITY_SUB_STRUCTURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_REFERENCE_FRAME',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_SYSTEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DERIVED_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DEVICE_TERMINAL_MAP',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FABRICATION_JOINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT_TERMINAL_DEFINITION',

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'GROUP_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'INTERCONNECT_MODULE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'INTERFACE_MOUNTED_JOIN',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'JOIN_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAND_TEMPLATE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAYER',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAYER_CONNECTION_POINT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'MOUNTING_RESTRICTION_AREA',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PACKAGE_BODY',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PACKAGE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PACKAGED_PART_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PART_CONNECTED_TERMINALS_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PART_TEMPLATE_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PASSAGE_TECHNOLOGY',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_CONNECTIVITY_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_CONNECTIVITY_ELEMENT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_NETWORK',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'POSITIONAL_BOUNDARY',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'POSITIONAL_BOUNDARY_MEMBER',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PRINTED_PART_CROSS_SECTION_TEMPLATE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PRINTED_PART_TEMPLATE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'SEQUENTIAL_LAMINATE_PASSAGE_BASED_FABRICATION_JOINT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'SHAPE_MODIFICATION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'STRATUM_CONCEPT_RELATIONSHIP',

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_SURFACE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TOLERANCE_ZONE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TOLERANCE_ZONE_BOUNDARY',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VIEWING_PLANE'] * TYPEOF(sa)) <= 1)) )) = 0);
WR2: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_BOND_DEFINITION',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_JOINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'ASSEMBLY_MODULE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MINIMALLY_DEFINED_BARE_DIE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_FUNCTIONAL_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_INTERFACE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'COMPONENT_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONNECTIVITY_SUB_STRUCTURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_REFERENCE_FRAME',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DATUM_SYSTEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'DEVICE_TERMINAL_MAP',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FABRICATION_JOINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'FUNCTIONAL_UNIT_TERMINAL_DEFINITION',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'GROUP_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERCONNECT_MODULE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'INTERFACE_MOUNTED_JOIN',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'JOIN_SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAND_TEMPLATE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LAYER',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

' LAYER_CONNECTION_POINT ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' MOUNTING_RESTRICTION_AREA ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PACKAGE_BODY ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PACKAGE_TERMINAL ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PACKAGED_PART_TERMINAL ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PART_CONNECTED_TERMINALS_DEFINITION ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PART_TEMPLATE_DEFINITION ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PASSAGE_TECHNOLOGY ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PHYSICAL_CONNECTIVITY_DEFINITION ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PHYSICAL_CONNECTIVITY_ELEMENT ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PHYSICAL_NETWORK ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PHYSICAL_UNIT_DATUM ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' POSITIONAL_BOUNDARY ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' POSITIONAL_BOUNDARY_MEMBER ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_CROSS_SECTION_TEMPLATE_TERMINAL ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' PRINTED_PART_TEMPLATE_TERMINAL ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SEQUENTIAL_LAMINATE_PASSAGE_BASED_FABRICATION_JOINT ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' SHAPE_MODIFICATION ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' STRATUM_CONCEPT_RELATIONSHIP ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' STRATUM_FEATURE ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' STRATUM_SURFACE ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' TOLERANCE_ZONE ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' TOLERANCE_ZONE_BOUNDARY ',
' ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
' VIEWING_PLANE' ] * TYPEOF(sa)) <= 1)) )) = 0);
WR3: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF([

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'ASSEMBLY_BOND_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'ASSEMBLY_JOINT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'COMPONENT_FUNCTIONAL_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'COMPONENT_INTERFACE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'COMPONENT_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'COMPONENT_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'CONNECTIVITY_SUB_STRUCTURE',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'DATUM_REFERENCE_FRAME',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'DATUM_SYSTEM',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'DERIVED_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'DEVICE_TERMINAL_MAP',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'FABRICATION_JOINT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'FUNCTIONAL_UNIT_TERMINAL_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'GROUP_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'INTERFACE_MOUNTED_JOIN',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'JOIN_SHAPE_ASPECT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAND_TEMPLATE_TERMINAL',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAYER',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'LAYER_CONNECTION_POINT',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'MOUNTING_RESTRICTION_AREA',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PACKAGE_BODY',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PART_CONNECTED_TERMINALS_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PART_TEMPLATE_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PASSAGE_TECHNOLOGY',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_CONNECTIVITY_DEFINITION',
 'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
 'PHYSICAL_CONNECTIVITY_ELEMENT',

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_NETWORK',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_TARGET',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITIONAL_BOUNDARY',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POSITIONAL_BOUNDARY_MEMBER',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_CROSS_SECTION_TEMPLATE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRINTED_PART_TEMPLATE_TERMINAL',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SEQUENTIAL_LAMINATE_PASSAGE_BASED_FABRICATION_JOINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_MODIFICATION',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_CONCEPT_RELATIONSHIP',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STRATUM_SURFACE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TOLERANCE_ZONE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TOLERANCE_ZONE_BOUNDARY',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'VIEWING_PLANE'] * TYPEOF(sa)) <= 1)) )) = 0);
WR4: (SIZEOF(QUERY ( sa <* shape_aspect | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_FEATURE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM_TARGET',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PHYSICAL_UNIT_DATUM'] * TYPEOF(sa)) <= 1)) )) = 0);

```

END_RULE; -- subtype_combination_shape_aspect

RULE subtype_exclusive_action_request_assignment FOR (
action_request_assignment);

WHERE

```

WR1: (SIZEOF(QUERY ( ara <* action_request_assignment | (NOT (SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CHANGE_REQUEST',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'START_REQUEST'] * TYPEOF(ara)) <= 1)) )) = 0);

```

```
END_RULE; -- subtype_exclusive_action_request_assignment
```

```
RULE subtype_exclusive_dimensional_location FOR (dimensional_location);
```

```
WHERE
```

```
  WR1: (SIZEOF(QUERY ( dl <* dimensional_location | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANGULAR_DIMENSION_WITH_ORIENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_LOCATION_WITH_DIRECTION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_LOCATION_WITH_PATH'] * TYPEOF(dl)) <= 1)) )) = 0);
```

```
END_RULE; -- subtype_exclusive_dimensional_location
```

```
RULE subtype_exclusive_dimensional_size FOR (dimensional_size);
```

```
WHERE
```

```
  WR1: (SIZEOF(QUERY ( ds <* dimensional_size | ((NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE_DIMENSION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANGULAR_SIZE'] * TYPEOF(ds)) <= 1)) OR (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE_DIMENSION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'OPPOSING_BOUNDARY_DIMENSIONAL_SIZE'] * TYPEOF(ds)) <= 1))) ))
    = 0);
```

```
END_RULE; -- subtype_exclusive_dimensional_size
```

```
RULE subtype_exclusive_mapped_item FOR (mapped_item);
```

```
WHERE
```

```
  WR1: (SIZEOF(QUERY ( mi <* mapped_item | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANNOTATION_TEXT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_MODULE_USAGE_VIEW_CONNECTOR_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ANNOTATION_SYMBOL'] * TYPEOF(mi)) <= 1)) )) = 0);
```

```
END_RULE; -- subtype_exclusive_mapped_item
```

```
RULE subtype_exclusive_product_definition FOR (product_definition);
```

WHERE

```
WR1: (SIZEOF(QUERY ( pd <* product_definition | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BUS_STRUCTURAL_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPONENT_FUNCTIONAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'NETWORK_NODE_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM'] * TYPEOF(pd)) <= 1)) )) = 0);
```

END_RULE; -- subtype_exclusive_product_definition

RULE subtype_exclusive_product_definition_relationship FOR (
 product_definition_relationship);

WHERE

```
WR1: (SIZEOF(QUERY ( pdr <* product_definition_relationship | (NOT (
    SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_USAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SUPPLIED_PART_RELATIONSHIP'] * TYPEOF(pdr)) <= 1)) )) = 0);
```

END_RULE; -- subtype_exclusive_product_definition_relationship

RULE subtype_exclusive_property_definition FOR (property_definition);

WHERE

```
WR1: (SIZEOF(QUERY ( pd <* property_definition | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_SIZE_PROPERTY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRIC_TOLERANCE_GROUP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_PROPERTY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_SHAPE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY',
```



```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SIGNAL'] * TYPEOF(pd)) <= 1)) )) = 0);

END_RULE; -- subtype_exclusive_property_definition

RULE subtype_exclusive_property_definition_representation FOR (
    property_definition_representation);

WHERE
    WR1: (SIZEOF(QUERY ( pdr <* property_definition_representation | (NOT
        (SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SHAPE_DEFINITION_REPRESENTATION',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'MATERIAL_PROPERTY_REPRESENTATION'] * TYPEOF(pdr)) <= 1)) )) =
        0));

END_RULE; -- subtype_exclusive_property_definition_representation

RULE subtype_exclusive_representation FOR (representation);

WHERE
    WR1: (SIZEOF(QUERY ( rep <* representation | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ANALYTICAL_MODEL_PORT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPONENT_LOCATION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_FUNCTION_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRUCTURED_TEXT_REPRESENTATION_ITEM',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TEXT_STRING_REPRESENTATION'] * TYPEOF(rep)) <= 1)) )) = 0));

END_RULE; -- subtype_exclusive_representation

RULE subtype_exclusive_representation_context FOR (
    representation_context);

WHERE
    WR1: (SIZEOF(QUERY ( rc <* representation_context | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'GEOMETRIC_REPRESENTATION_CONTEXT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PARAMETRIC_REPRESENTATION_CONTEXT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRUCTURED_TEXT_REPRESENTATION_CONTEXT'] * TYPEOF(rc)) <= 1)) ))
        = 0);

```

```
END_RULE; -- subtype_exclusive_representation_context
```

```
RULE subtype_exclusive_shape_aspect_relationship FOR (
    shape_aspect_relationship);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( sar <* shape_aspect_relationship | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_JOINT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DEVICE_TERMINAL_MAP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIMENSIONAL_LOCATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'INTERFACE_MOUNTED_JOIN',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAKE_FROM_CONNECTIVITY_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAKE_FROM_FEATURE_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_ELEMENT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_DERIVING_RELATIONSHIP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'STRATUM_CONCEPT_RELATIONSHIP'] * TYPEOF(sar)) <= 1)) )) = 0);
```

```
END_RULE; -- subtype_exclusive_shape_aspect_relationship
```

```
RULE subtype_mandatory_action FOR (action);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( act <* action | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DIRECTED_ACTION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXECUTED_ACTION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_ACTION'] * TYPEOF(act)) = 1)) )) = 0);
```

```
END_RULE; -- subtype_mandatory_action
```

```
RULE subtype_mandatory_address FOR (address);
```

```
WHERE
```

```
WR1: (SIZEOF(QUERY ( add <* address | (NOT (SIZEOF([
```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PERSONAL_ADDRESS',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ORGANIZATIONAL_ADDRESS'] * TYPEOF(add)) = 1)) )) = 0);

END_RULE; -- subtype_mandatory_address

RULE subtype_mandatory_characterized_object FOR (characterized_object);

WHERE
    WR1: (SIZEOF(QUERY ( co <* characterized_object | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CHARACTERIZED_PRODUCT_CATEGORY',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'DESIGN_OBJECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'EXTERNAL_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RULE_DEFINITION',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SIGNAL',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'STRATUM_TECHNOLOGY'] * TYPEOF(co)) = 1)) )) = 0);

END_RULE; -- subtype_mandatory_characterized_object

RULE subtype_mandatory_colour FOR (colour);

WHERE
    WR1: (SIZEOF(QUERY ( c <* colour | (NOT ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COLOUR_RGB') IN TYPEOF(c))) )) = 0);

END_RULE; -- subtype_mandatory_colour

RULE subtype_mandatory_composite_shape_aspect FOR (
    composite_shape_aspect);

WHERE
    WR1: (SIZEOF(QUERY ( csa <* composite_shape_aspect | (NOT (SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_GROUP_SHAPE_ASPECT',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_UNIT_SHAPE_ASPECT'] * TYPEOF(csa)) = 1)) )) = 0);

END_RULE; -- subtype_mandatory_composite_shape_aspect

```

```

RULE subtype_mandatory_date FOR (date);

WHERE
  WR1: (SIZEOF(QUERY ( d <* date | (NOT ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CALENDAR_DATE') IN TYPEOF(d))) )) = 0);

END_RULE; -- subtype_mandatory_date


RULE subtype_mandatory_externally_defined_item FOR (
  externally_defined_item);

WHERE
  WR1: (SIZEOF(QUERY ( edi <* externally_defined_item | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOND_CATEGORY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNAL_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_CURVE_FONT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_FUNCTIONAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_HATCH_STYLE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_PHYSICAL_UNIT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_REPRESENTATION_ITEM',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_SYMBOL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_TEXT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_TILE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EXTERNALLY_DEFINED_TILE_STYLE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LIBRARY_DEFINED_MODEL',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PART_TEMPLATE_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_DEFINITION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SIGNAL_CATEGORY'] * TYPEOF(edi)) = 1)) )) = 0);

END_RULE; -- subtype_mandatory_externally_defined_item


RULE subtype_mandatory_geometric_tolerance FOR (geometric_tolerance);

```

WHERE

```
WR1: (SIZEOF(QUERY ( gt <* geometric_tolerance | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_UNIT_GEOMETRIC_TOLERANCE'] * TYPEOF(gt)) = 1)) )) =
    0);
```

END_RULE; -- subtype_mandatory_geometric_tolerance

RULE subtype_mandatory_group FOR (group);

WHERE

```
WR1: (SIZEOF(QUERY ( grp <* group | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOND_CATEGORY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FUNCTIONAL_TERMINAL_GROUP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GROUPED_REQUIREMENTS_PROPERTY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MATERIAL_ELECTRICAL_CONDUCTIVITY_CATEGORY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PARAMETER_TYPE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRINTED_PART_TEMPLATE_TERMINAL_CONNECTION_ZONE_CATEGORY',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENT_ALLOCATION_GROUP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REQUIREMENTS_PROPERTY_GROUP',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RULE_SET',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SIGNAL_CATEGORY'] * TYPEOF(grp)) = 1)) )) = 0);
```

END_RULE; -- subtype_mandatory_group

RULE subtype_mandatory_pre_defined_item FOR (pre_defined_item);

WHERE

```
WR1: (SIZEOF(QUERY ( pdi <* pre_defined_item | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRE_DEFINED_TEXT_FONT',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRE_DEFINED_CURVE_FONT'] * TYPEOF(pdi)) = 1)) )) = 0);
```

END_RULE; -- subtype_mandatory_pre_defined_item

RULE subtype_mandatory_product_definition_usage FOR (
 product_definition_usage);

WHERE

```
WR1: (SIZEOF(QUERY ( pdu <* product_definition_usage | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ASSEMBLY_COMPONENT_USAGE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAKE_FROM_USAGE_OPTION'] * TYPEOF(pdu)) = 1)) )) = 0);
```

END_RULE; -- subtype_mandatory_product_definition_usage

RULE subtype_mandatory_runout_zone_orientation FOR (
 runout_zone_orientation);

WHERE

```
WR1: (SIZEOF(QUERY ( rzo <* runout_zone_orientation | (NOT (SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION'] * TYPEOF(rzo))
    = 1)) )) = 0);
```

END_RULE; -- subtype_mandatory_runout_zone_orientation

RULE subtype_mandatory_shape_representation FOR (shape_representation);

WHERE

```
WR1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT ((SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ADVANCED_BREP_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CSG_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACETED_BREP_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_DIMENSION_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION'] * TYPEOF(sr)) =
    1) OR (SIZEOF(QUERY ( it <* sr\representation.items | (NOT (
```

```

SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_2D',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'AXIS2_PLACEMENT_3D',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CARTESIAN_POINT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'MAPPED_ITEM',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'STYLED_ITEM'] * TYPEOF(it)) = 1)) )) = 0) OR (SIZEOF(
QUERY ( pdr <* USEDIN(sr,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.' + 'USED_REPRESENTATION')
| (NOT ((SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PROPERTY_DEFINITION_REPRESENTATION.',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_DEFINITION_REPRESENTATION'] * TYPEOF(pdr)) >= 1) AND (
SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SHAPE_ASPECT_RELATIONSHIP'] * TYPEOF(pdr.definition)) >= 1)))) ))
= 0))) )) = 0);

```

END_RULE; -- subtype_mandatory_shape_representation

RULE subtype_mandatory_text_literal FOR (text_literal);

WHERE

```

WR1: (SIZEOF(QUERY ( tl <* text_literal | (NOT (((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEXT_LITERAL_WITH_EXTENT') IN TYPEOF(tl)) OR ((
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TEXT_LITERAL_WITH_ASSOCIATED_CURVES') IN TYPEOF(tl)))))) )) = 0);

```

END_RULE; -- subtype_mandatory_text_literal

RULE supplied_product_version_unique_constraint FOR (
product_definition_formation_with_specified_source);

LOCAL

```

aoa_bag : BAG OF applied_organization_assignment;
spv      : BAG OF product_definition_formation_with_specified_source
:= QUERY ( r <*
product_definition_formation_with_specified_source | (
SIZEOF(QUERY ( pd <* USEDIN(r,
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PRODUCT_DEFINITION.FORMATION') | (SIZEOF(QUERY ( pdr <*

```

```

        USEDIN(pd,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION')
        | ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SUPPLIED_PART_RELATIONSHIP') IN TYPEOF(pdr)) )) > 0) ))
    > 0) );
    spn_bag : BAG OF STRING;
    o_bag   : BAG OF organization := [];
    pass    : BOOLEAN := TRUE;
    pdf_bag : BAG OF product_definition_formation_with_specified_source;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(spvr) BY 1;
    aoa_bag := QUERY ( aoa <* USEDIN(spvr[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (aoa.role.name =
    'supplier') );
    REPEAT j := 1 TO SIZEOF(aoa_bag) BY 1;
        IF EXISTS(aoa_bag[j].assigned_organization) THEN
            IF NOT (aoa_bag[j].assigned_organization IN o_bag) THEN
                o_bag := o_bag + aoa_bag[j].assigned_organization;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(o_bag) BY 1;
    IF NOT pass THEN
        ESCAPE;
    END_IF;
    pdf_bag := QUERY ( pdf <* spvr | (SIZEOF(QUERY ( aoa <* USEDIN(pdf,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | ((aoa.role.name =
    'supplier') AND (aoa.assigned_organization :=: o_bag[i])) )) > 0) ));
    spn_bag := [];
    REPEAT j := 1 TO SIZEOF(pdf_bag) BY 1;
        IF EXISTS(pdf_bag[j].of_product.id) THEN
            IF pdf_bag[j].of_product.id IN spn_bag THEN
                pass := FALSE;
                ESCAPE;
            ELSE
                spn_bag := spn_bag + pdf_bag[j].of_product.id;
            END_IF;
        END_IF;
    END_REPEAT;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- supplied_product_version_unique_constraint

```



```

RULE topological_junction_unique_constraint FOR (shape_aspect);

LOCAL
  name_bag : BAG OF STRING;
  sar_bag  : BAG OF shape_aspect_relationship;
  s_bag    : BAG OF shape_aspect := [];
  tj       : BAG OF shape_aspect := QUERY ( r <* shape_aspect | (r.
      description = 'topological junction') );
  pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(tj) BY 1;
  sar_bag := QUERY ( sar <* USEDIN(tj[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | ((sar.name =
    'topological junction scope') AND ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PHYSICAL_CONNECTIVITY_DEFINITION') IN TYPEOF(sar.
    relating_shape_aspect))) );
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].relating_shape_aspect) THEN
      IF NOT (sar_bag[j].relating_shape_aspect IN s_bag) THEN
        s_bag := s_bag + sar_bag[j].relating_shape_aspect;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO SIZEOF(s_bag) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  sar_bag := QUERY ( sar <* USEDIN(s_bag[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.name =
    'topological junction scope') );
  name_bag := [];
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].related_shape_aspect.name) THEN
      IF sar_bag[j].related_shape_aspect.name IN name_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        name_bag := name_bag + sar_bag[j].related_shape_aspect.name;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- topological_junction_unique_constraint

```

```
RULE transmission_line_functional_unit_terminal_allocation_unique_constraint
  FOR ( shape_aspect_relationship);
```

```
  LOCAL
```

```
    sar_bag : BAG OF shape_aspect_relationship;
    ut_bag  : BAG OF shape_aspect := [];
    tlfuta  : BAG OF shape_aspect_relationship := QUERY ( r <*
      shape_aspect_relationship | (r.name =
        'transmission line functional terminal allocation') );
    pass    : BOOLEAN := TRUE;
    sa_bag  : BAG OF shape_aspect;
```

```
  END_LOCAL;
```

```
  REPEAT i := 1 TO SIZEOF(tlfuta) BY 1;
```

```
    IF EXISTS(tlfuta[i].relating_shape_aspect) THEN
```

```
      IF tlfuta[i].relating_shape_aspect.description = 'unit terminal'
      THEN
```

```
        IF NOT (tlfuta[i].relating_shape_aspect IN ut_bag) THEN
          ut_bag := ut_bag + tlfuta[i].relating_shape_aspect;
```

```
        END_IF;
```

```
      END_IF;
```

```
    END_IF;
```

```
  END_REPEAT;
```

```
  REPEAT i := 1 TO SIZEOF(ut_bag) BY 1;
```

```
    IF NOT pass THEN
```

```
      ESCAPE;
```

```
    END_IF;
```

```
    sar_bag := QUERY ( sar <* tlfuta | ((sar.related_shape_aspect.
      description = 'printed component join terminal') AND (sar.
      relating_shape_aspect :=: ut_bag[i])) );
```

```
    sa_bag := [];
```

```
    REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
```

```
      IF EXISTS(sar_bag[j].related_shape_aspect) THEN
```

```
        IF sar_bag[j].related_shape_aspect IN sa_bag THEN
```

```
          pass := FALSE;
```

```
          ESCAPE;
```

```
        ELSE
```

```
          sa_bag := sa_bag + sar_bag[j].related_shape_aspect;
```

```
        END_IF;
```

```
      END_IF;
```

```
    END_REPEAT;
```

```
  END_REPEAT;
```

```
WHERE
```

```
  WR1: pass;
```

```
END_RULE;
```

```
-- transmission_line_functional_unit_terminal_allocation_unique_constraint
```

```
RULE versioned_action_request_requires_approval FOR (
  versioned_action_request, applied_approval_assignment);
```

```

WHERE
  WR1: (SIZEOF(QUERY ( varq <* versioned_action_request | (NOT (SIZEOF(
    QUERY ( aaa <* applied_approval_assignment | (varq IN aaa.
      items) )) = 1)) )) = 0));

END_RULE; -- versioned_action_request_requires_approval


RULE versioned_action_request_requires_date_or_date_and_time FOR (
  versioned_action_request, applied_date_and_time_assignment,
  applied_date_assignment);

WHERE
  WR1: (SIZEOF(QUERY ( varq <* versioned_action_request | (NOT ((SIZEOF(
    QUERY ( adata <* applied_date_and_time_assignment | (varq IN
      adata.items) )) = 1) OR (SIZEOF(QUERY ( ada <*
        applied_date_assignment | (varq IN ada.items) )) = 1)))) )) = 0));

END_RULE; -- versioned_action_request_requires_date_or_date_and_time


RULE versioned_action_request_requires_person_organization FOR (
  versioned_action_request,
  applied_person_and_organization_assignment,
  applied_organization_assignment);

WHERE
  WR1: (SIZEOF(QUERY ( varq <* versioned_action_request | (NOT ((SIZEOF(
    QUERY ( apaoa <* applied_person_and_organization_assignment |
      (varq IN apaoa.items) )) >= 1) OR (SIZEOF(QUERY ( aoa <*
        applied_organization_assignment | (varq IN aoa.items) )) >= 1)))) ))
    = 0));

END_RULE; -- versioned_action_request_requires_person_organization


RULE versioned_action_request_requires_status FOR (
  versioned_action_request, action_request_status);

WHERE
  WR1: (SIZEOF(QUERY ( ar <* versioned_action_request | (NOT (SIZEOF(
    QUERY ( ars <* action_request_status | (ar ::= ars.
      assigned_request) )) = 1)) )) = 0));

END_RULE; -- versioned_action_request_requires_status


RULE via_template_terminal_unique_constraint FOR (shape_aspect);

LOCAL

```

```

name_bag : BAG OF STRING;
vt       : BAG OF shape_aspect := QUERY ( r <* shape_aspect | (r.
      description = 'via template') );
sar_bag  : BAG OF shape_aspect_relationship;
pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(vt) BY 1;
  IF NOT pass THEN
    ESCAPE;
  END_IF;
  sar_bag := QUERY ( sar <* USEDIN(vt[i],
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((sar.name =
    'access mechanism') AND (sar.related_shape_aspect.description =
    'via template terminal')) );
  name_bag := [];
  REPEAT j := 1 TO SIZEOF(sar_bag) BY 1;
    IF EXISTS(sar_bag[j].related_shape_aspect.name) THEN
      IF sar_bag[j].related_shape_aspect.name IN name_bag THEN
        pass := FALSE;
        ESCAPE;
      ELSE
        name_bag := name_bag + sar_bag[j].related_shape_aspect.name;
      END_IF;
    END_IF;
  END_REPEAT;
END_REPEAT;

WHERE
  WR1: pass;

END_RULE; -- via_template_terminal_unique_constraint

```

```

RULE work_order_unique_constraint FOR (directed_action);

```

```

LOCAL
  name_bag : BAG OF STRING := [];
  r        : BAG OF directed_action := QUERY ( r <* directed_action |
    (SIZEOF(QUERY ( aa <* USEDIN(r,
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'ACTION_ASSIGNMENT.ASSIGNED_ACTION') | (((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'CHANGE') IN TYPEOF(aa)) OR ((
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'START_WORK') IN TYPEOF(aa))) ) > 0) );
  pass     : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
  IF EXISTS(r[i].directive.name) THEN
    IF r[i].directive.name IN name_bag THEN
      pass := FALSE;
    END_IF;
  END_REPEAT;

```

```

        ESCAPE;
    ELSE
        name_bag := name_bag + r[i].directive.name;
    END_IF;
END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- work_order_unique_constraint

RULE work_request_unique_constraint FOR (versioned_action_request);

LOCAL
    r      : BAG OF versioned_action_request := QUERY ( r <*
        versioned_action_request | (SIZEOF(QUERY ( ara <* USEDIN(r,
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') | (((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CHANGE_REQUEST') IN TYPEOF(ara)) OR ((
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'START_REQUEST') IN TYPEOF(ara))) )) > 0) );
    pass    : BOOLEAN := TRUE;
    id_bag  : BAG OF STRING := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(r) BY 1;
    IF EXISTS(r[i].id) THEN
        IF r[i].id IN id_bag THEN
            pass := FALSE;
            ESCAPE;
        ELSE
            id_bag := id_bag + r[i].id;
        END_IF;
    END_IF;
END_REPEAT;

WHERE
    WR1: pass;

END_RULE; -- work_request_unique_constraint

FUNCTION acyclic_curve_replica(
    rep: curve_replica;
    parent: curve
): BOOLEAN;
IF NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE_REPLICA'
    IN TYPEOF(parent)) THEN

```

```

        RETURN(TRUE);
    END_IF;
    IF parent :=: rep THEN
        RETURN(FALSE);
    ELSE
        RETURN(acyclic_curve_replica(rep,parent\curve_replica.parent_curve));
    END_IF;

END_FUNCTION; -- acyclic_curve_replica

FUNCTION acyclic_mapped_representation(
    parent_set: SET OF representation;
    children_set: SET OF representation_item
): BOOLEAN;

LOCAL
    i : INTEGER;
    x : SET OF representation_item;
    y : SET OF representation_item;
END_LOCAL;
x := QUERY ( z <* children_set | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MAPPED_ITEM'
    IN TYPEOF(z)) );
IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
        IF x[i]\mapped_item.mapping_source.mapped_representation IN
            parent_set THEN
            RETURN(FALSE);
        END_IF;
        IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
            .mapping_source.mapped_representation,x[i]\mapped_item.
            mapping_source.mapped_representation.items) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_IF;
x := children_set - x;
IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
        y := QUERY ( z <* bag_to_set(USEDIN(x[i],'')) | (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'REPRESENTATION_ITEM'
            IN TYPEOF(z)) );
        IF NOT acyclic_mapped_representation(parent_set,y) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_IF;
RETURN(TRUE);

```

```
END_FUNCTION; -- acyclic_mapped_representation
```

```
FUNCTION acyclic_point_replica(
    rep: point_replica;
    parent: point
): BOOLEAN;
IF NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POINT_REPLICA'
    IN TYPEOF(parent)) THEN
    RETURN(TRUE);
END_IF;
IF parent :=: rep THEN
    RETURN(FALSE);
ELSE
    RETURN(acyclic_point_replica(rep, parent\point_replica.parent_pt));
END_IF;
```

```
END_FUNCTION; -- acyclic_point_replica
```

```
FUNCTION acyclic_product_category_relationship(
    relation: product_category_relationship;
    children: SET OF product_category
): LOGICAL;

LOCAL
    i          : INTEGER;
    x          : SET OF product_category_relationship;
    local_children : SET OF product_category;
END_LOCAL;
REPEAT i := 1 TO HIINDEX(children) BY 1;
    IF relation.category :=: children[i] THEN
        RETURN(FALSE);
    END_IF;
END_REPEAT;
x := bag_to_set(USEDIN(relation.category,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));
local_children := children + relation.category;
IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
        IF NOT acyclic_product_category_relationship(x[i], local_children)
            THEN
                RETURN(FALSE);
            END_IF;
    END_REPEAT;
END_IF;
RETURN(TRUE);
```

```
END_FUNCTION; -- acyclic_product_category_relationship
```

```

FUNCTION acyclic_product_definition_relationship(
    relation: product_definition_relationship;
    relatives: SET [1:?] OF product_definition;
    specific_relation: STRING
): LOGICAL;

LOCAL
    x : SET OF product_definition_relationship;
END_LOCAL;
IF relation.relating_product_definition IN relatives THEN
    RETURN(FALSE);
END_IF;
x := QUERY ( pd <* bag_to_set(USEDIN(relation.
    relating_product_definition,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION'))
    | (specific_relation IN TYPEOF(pd)) );
REPEAT i := 1 TO HIINDEX(x) BY 1;
    IF NOT acyclic_product_definition_relationship(x[i],relatives +
        relation.relating_product_definition,specific_relation) THEN
        RETURN(FALSE);
    END_IF;
END_REPEAT;
RETURN(TRUE);

END_FUNCTION; -- acyclic_product_definition_relationship

```

```

FUNCTION acyclic_solid_replica(
    rep: solid_replica;
    parent: solid_model
): BOOLEAN;
IF NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SOLID_REPLICA'
    IN TYPEOF(parent)) THEN
    RETURN(TRUE);
END_IF;
IF parent :=: rep THEN
    RETURN(FALSE);
ELSE
    RETURN(acyclic_solid_replica(rep,parent\solid_replica.parent_solid));
END_IF;

END_FUNCTION; -- acyclic_solid_replica

```

```

FUNCTION acyclic_surface_replica(
    rep: surface_replica;
    parent: surface
): BOOLEAN;

```



```

IF NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE_REPLICA'
    IN TYPEOF(parent)) THEN
    RETURN(TRUE);
END_IF;
IF parent ::= rep THEN
    RETURN(FALSE);
ELSE
    RETURN(acyclic_surface_replica(rep,parent\surface_replica.
        parent_surface));
END_IF;

END_FUNCTION; -- acyclic_surface_replica

FUNCTION applied_date_correlation(
    e: applied_date_assignment;
    schema_name: STRING
): BOOLEAN;

LOCAL
    d_role : STRING;
END_LOCAL;
d_role := e\date_assignment.role.name;
CASE d_role OF
    'creation date' : IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name + '.PRODUCT_DEFINITION')
            IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'request date' : IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name +
            '.VERSIONED_ACTION_REQUEST') IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'release date' : IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | (SIZEOF([schema_name + '.CHANGE',
            schema_name + '.START_WORK'] * TYPEOF(x)) = 1) )) THEN
        RETURN(FALSE);
    END_IF;
    'start date' : IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | (SIZEOF([schema_name + '.CHANGE',
            schema_name + '.START_WORK'] * TYPEOF(x)) = 1) )) THEN
        RETURN(FALSE);
    END_IF;
    'sign off date' : IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name +
            '.APPROVAL_PERSON_ORGANIZATION') IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'contract date' : IF SIZEOF(e.items) <> SIZEOF(

```

```

        QUERY ( x <* e.items | ((schema_name +
        '.CONTRACT') IN TYPEOF(x)) ))
        THEN
        RETURN(FALSE);
    END_IF;
    'certification date' :          IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name + '.CERTIFICATION') IN
        TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'classification date' :        IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name +
        '.SECURITY_CLASSIFICATION') IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'declassification date' :      IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name +
        '.SECURITY_CLASSIFICATION') IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    OTHERWISE                :      RETURN(TRUE);
END_CASE;
RETURN(TRUE);

```

```
END_FUNCTION; -- applied_date_correlation
```

```

FUNCTION applied_date_time_correlation(
    e: applied_date_and_time_assignment;
    schema_name: STRING
): BOOLEAN;

LOCAL
    dt_role : STRING;
END_LOCAL;
dt_role := e\date_and_time_assignment.role.name;
CASE dt_role OF
    'creation date' :          IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name + '.PRODUCT_DEFINITION')
        IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'request date' :          IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | ((schema_name +
        '.VERSIONED_ACTION_REQUEST') IN TYPEOF(x)) )) THEN
        RETURN(FALSE);
    END_IF;
    'release date' :          IF SIZEOF(e.items) <> SIZEOF(
        QUERY ( x <* e.items | (SIZEOF([schema_name + '.CHANGE',
        schema_name + '.START_WORK'] * TYPEOF(x)) = 1) )) THEN
        RETURN(FALSE);
    END_IF;

```

```

'start date'          :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | (SIZEOF([schema_name + '.CHANGE',
    schema_name + '.START_WORK'] * TYPEOF(x)) = 1) )) THEN
    RETURN(FALSE);
END_IF;
'sign off date'       :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name +
    '.APPROVAL_PERSON_ORGANIZATION') IN TYPEOF(x)) )) THEN
    RETURN(FALSE);
END_IF;
'contract date'       :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name + '.CONTRACT') IN TYPEOF(x)) ))
    THEN
    RETURN(FALSE);
END_IF;
'certification date'  :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name + '.CERTIFICATION') IN
    TYPEOF(x)) )) THEN
    RETURN(FALSE);
END_IF;
'classification date'  :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name +
    '.SECURITY_CLASSIFICATION') IN TYPEOF(x)) )) THEN
    RETURN(FALSE);
END_IF;
'declassification date' :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name +
    '.SECURITY_CLASSIFICATION') IN TYPEOF(x)) )) THEN
    RETURN(FALSE);
END_IF;
'participant date and time' :          IF SIZEOF(e.items) <> SIZEOF(
    QUERY ( x < * e.items | ((schema_name + '.RULE_ACTION') IN
    TYPEOF(x)) )) THEN
    RETURN(FALSE);
END_IF;
OTHERWISE              :          RETURN(TRUE);
END_CASE;
RETURN(TRUE);

```

```
END_FUNCTION; -- applied_date_time_correlation
```

```

FUNCTION assembly_shape_is_defined(
    assy: next_assembly_usage_occurrence;
    schma: STRING
): BOOLEAN;

```

```

LOCAL
    srr_set    : SET OF shape_representation_relationship := [];
    i          : INTEGER;
    j          : INTEGER;
    sdr_set    : SET OF shape_definition_representation := [];

```

```

pr1_set    : SET OF property_definition := [];
pdrel_set  : SET OF product_definition_relationship := [];
pr2_set    : SET OF property_definition := [];
END_LOCAL;
pr1_set := bag_to_set(USEDIN(assy.related_product_definition,schma +
'.PROPERTY_DEFINITION.DEFINITION'));
REPEAT i := 1 TO HIINDEX(pr1_set) BY 1;
    sdr_set := sdr_set + QUERY ( pdr <* USEDIN(pr1_set[i],schma +
'.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((schma +
'.SHAPE_DEFINITION_REPRESENTATION') IN TYPEOF(pdr)) );
END_REPEAT;
pdrel_set := bag_to_set(USEDIN(assy.related_product_definition,schma +
'.PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION'));
REPEAT j := 1 TO HIINDEX(pdrel_set) BY 1;
    pr2_set := pr2_set + USEDIN(pdrel_set[j],schma +
'.PROPERTY_DEFINITION.DEFINITION');
END_REPEAT;
REPEAT i := 1 TO HIINDEX(pr2_set) BY 1;
    sdr_set := sdr_set + QUERY ( pdr <* USEDIN(pr2_set[i],schma +
'.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((schma +
'.SHAPE_DEFINITION_REPRESENTATION') IN TYPEOF(pdr)) );
END_REPEAT;
IF SIZEOF(sdr_set) > 0 THEN
    REPEAT i := 1 TO HIINDEX(sdr_set) BY 1;
        srr_set := QUERY ( rr <* bag_to_set(USEDIN(sdr_set[i]\
property_definition_representation.used_representation,schma +
'.REPRESENTATION_RELATIONSHIP.REP_2')) | ((schma +
'.SHAPE_REPRESENTATION_RELATIONSHIP') IN TYPEOF(rr)) );
        IF SIZEOF(srr_set) > 0 THEN
            REPEAT j := 1 TO HIINDEX(srr_set) BY 1;
                IF SIZEOF(QUERY ( pdr <* bag_to_set(USEDIN(srr_set[j]\
representation_relationship.rep_1,schma +
'.PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION'))
| ((schma + '.SHAPE_DEFINITION_REPRESENTATION') IN TYPEOF(
pdr)) ) * QUERY ( pdr <* bag_to_set(USEDIN(assy.
relating_product_definition,schma +
'.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
schma + '.SHAPE_DEFINITION_REPRESENTATION') IN TYPEOF(pdr)) ))
>= 1 THEN
                    IF SIZEOF(QUERY ( cdsr <* USEDIN(srr_set[j],schma +
'.CONTEXT_DEPENDENT_SHAPE_REPRESENTATION.' +
'REPRESENTATION_RELATION') | (NOT (cdsr\
context_dependent_shape_representation.
represented_product_relation\property_definition.
definition :=: assy)) )) > 0 THEN
                        RETURN(FALSE);
                    END_IF;
                END_IF;
            END_REPEAT;
        END_IF;
    END_REPEAT;
END_IF;

```

```

    RETURN(TRUE);

END_FUNCTION; -- assembly_shape_is_defined


FUNCTION associated_surface(
    arg: pcurve_or_surface
): surface;

LOCAL
    surf : surface;
END_LOCAL;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PCURVE' IN
    TYPEOF(arg) THEN
    surf := arg.basis_surface;
ELSE
    surf := arg;
END_IF;
RETURN(surf);

END_FUNCTION; -- associated_surface


FUNCTION bag_to_set(
    the_bag: BAG OF GENERIC:intype
): SET OF GENERIC:intype;

LOCAL
    i      : INTEGER;
    the_set : SET OF GENERIC:intype := [];
END_LOCAL;
IF SIZEOF(the_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(the_bag) BY 1;
        the_set := the_set + the_bag[i];
    END_REPEAT;
END_IF;
RETURN(the_set);

END_FUNCTION; -- bag_to_set


FUNCTION base_axis(
    dim: INTEGER;
    axis1, axis2, axis3: direction
): LIST [2:3] OF direction;

LOCAL
    u      : LIST [2:3] OF direction;
    vec    : direction;

```

```

    factor : REAL;
END_LOCAL;
IF dim = 3 THEN
    u[3] := NVL(normalise(axis3), dummy_gri || direction([0,0,1]));
    u[1] := first_proj_axis(u[3], axis1);
    u[2] := second_proj_axis(u[3], u[1], axis2);
ELSE
    u[3] := ?;
    IF EXISTS(axis1) THEN
        u[1] := normalise(axis1);
        u[2] := orthogonal_complement(u[1]);
        IF EXISTS(axis2) THEN
            factor := dot_product(axis2, u[2]);
            IF factor < 0 THEN
                u[2].direction_ratios[1] := -u[2].direction_ratios[1];
                u[2].direction_ratios[2] := -u[2].direction_ratios[2];
            END_IF;
        END_IF;
    ELSE
        IF EXISTS(axis2) THEN
            u[2] := normalise(axis2);
            u[1] := orthogonal_complement(u[2]);
            u[1].direction_ratios[1] := -u[1].direction_ratios[1];
            u[1].direction_ratios[2] := -u[1].direction_ratios[2];
        ELSE
            u[1].name := '';
            u[2].name := '';
            u[1].direction_ratios[1] := 1;
            u[1].direction_ratios[2] := 0;
            u[2].direction_ratios[1] := 0;
            u[2].direction_ratios[2] := 1;
        END_IF;
    END_IF;
END_IF;
RETURN(u);

END_FUNCTION; -- base_axis

FUNCTION boolean_choose(
    b: BOOLEAN;
    choice1, choice2: GENERIC:item
): GENERIC:item;
IF b THEN
    RETURN(choice1);
ELSE
    RETURN(choice2);
END_IF;

END_FUNCTION; -- boolean_choose

```

```

FUNCTION build_2axes(
    ref_direction: direction
): LIST [2:2] OF direction;

    LOCAL
        u : LIST [2:2] OF direction;
    END_LOCAL;
    u[1] := NVL(normalise(ref_direction), dummy_gri || direction([1,0]));
    u[2] := orthogonal_complement(u[1]);
    RETURN(u);

END_FUNCTION; -- build_2axes


FUNCTION build_axes(
    axis, ref_direction: direction
): LIST [3:3] OF direction;

    LOCAL
        d : direction;
        u : LIST [3:3] OF direction;
    END_LOCAL;
    d := NVL(normalise(axis), dummy_gri || direction([0,0,1]));
    u[1] := first_proj_axis(d, ref_direction);
    u[2] := normalise(cross_product(d, u[1])).orientation;
    u[3] := d;
    RETURN(u);

END_FUNCTION; -- build_axes


FUNCTION conditional_reverse(
    p: BOOLEAN;
    an_item: reversible_topology
): reversible_topology;
    IF p THEN
        RETURN(an_item);
    ELSE
        RETURN(topology_reversed(an_item));
    END_IF;

END_FUNCTION; -- conditional_reverse


FUNCTION constraints_composite_curve_on_surface(
    c: composite_curve_on_surface
): BOOLEAN;

    LOCAL
        n_segments : INTEGER := SIZEOF(c.segments);
    END_LOCAL;

```

```

REPEAT k := 1 TO n_segments BY 1;
  IF (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PCURVE' IN
      TYPEOF(c\composite_curve.segments[k].parent_curve))) AND (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SURFACE_CURVE'
      IN TYPEOF(c\composite_curve.segments[k].parent_curve))) AND (NOT
    (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPOSITE_CURVE_ON_SURFACE'
      IN TYPEOF(c\composite_curve.segments[k].parent_curve))) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);

END_FUNCTION; -- constraints_composite_curve_on_surface

FUNCTION constraints_geometry_shell_based_surface_model(
  m: shell_based_surface_model
): BOOLEAN;

LOCAL
  result : BOOLEAN := TRUE;
END_LOCAL;
REPEAT j := 1 TO SIZEOF(m.sbsm_boundary) BY 1;
  IF (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'OPEN_SHELL'
      IN TYPEOF(m.sbsm_boundary[j]))) AND (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CLOSED_SHELL'
      IN TYPEOF(m.sbsm_boundary[j]))) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
END_REPEAT;
RETURN(result);

END_FUNCTION; -- constraints_geometry_shell_based_surface_model

FUNCTION constraints_geometry_shell_based_wireframe_model(
  m: shell_based_wireframe_model
): BOOLEAN;

LOCAL
  result : BOOLEAN := TRUE;
END_LOCAL;

```



```

REPEAT j := 1 TO SIZEOF(m.sbwmm_boundary) BY 1;
  IF (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'WIRE_SHELL'
    IN TYPEOF(m.sbwmm_boundary[j]))) AND (NOT (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VERTEX_SHELL'
    IN TYPEOF(m.sbwmm_boundary[j]))) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
END_REPEAT;
RETURN(result);

END_FUNCTION; -- constraints_geometry_shell_based_wireframe_model

FUNCTION constraints_param_b_spline(
  degree, up_knots, up_cp: INTEGER;
  knot_mult: LIST OF INTEGER;
  knots: LIST OF parameter_value
): BOOLEAN;

LOCAL
  k      : INTEGER;
  l      : INTEGER;
  sum    : INTEGER;
  result : BOOLEAN := TRUE;
END_LOCAL;
sum := knot_mult[1];
REPEAT i := 2 TO up_knots BY 1;
  sum := sum + knot_mult[i];
END_REPEAT;
IF (degree < 1) OR (up_knots < 2) OR (up_cp < degree) OR (sum <> (
  degree + up_cp + 2)) THEN
  result := FALSE;
  RETURN(result);
END_IF;
k := knot_mult[1];
IF (k < 1) OR (k > (degree + 1)) THEN
  result := FALSE;
  RETURN(result);
END_IF;
REPEAT i := 2 TO up_knots BY 1;
  IF (knot_mult[i] < 1) OR (knots[i] <= knots[i - 1]) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  k := knot_mult[i];
  IF (i < up_knots) AND (k > degree) THEN
    result := FALSE;
    RETURN(result);
  
```

```

        END_IF;
        IF (i = up_knots) AND (k > (degree + 1)) THEN
            result := FALSE;
            RETURN(result);
        END_IF;
    END_REPEAT;
    RETURN(result);

END_FUNCTION; -- constraints_param_b_spline

FUNCTION constraints_rectangular_composite_surface(
    s: rectangular_composite_surface
): BOOLEAN;
REPEAT i := 1 TO s.n_u BY 1;
    REPEAT j := 1 TO s.n_v BY 1;
        IF NOT ((
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'B_SPLINE_SURFACE'
            IN TYPEOF(s.segments[i][j].parent_surface)) OR (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'RECTANGULAR_TRIMMED_SURFACE'
            IN TYPEOF(s.segments[i][j].parent_surface))) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO s.n_u - 1 BY 1;
    REPEAT j := 1 TO s.n_v BY 1;
        IF s.segments[i][j].u_transition = discontinuous THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_REPEAT;
REPEAT i := 1 TO s.n_u BY 1;
    REPEAT j := 1 TO s.n_v - 1 BY 1;
        IF s.segments[i][j].v_transition = discontinuous THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_REPEAT;
RETURN(TRUE);

END_FUNCTION; -- constraints_rectangular_composite_surface

FUNCTION cross_product(
    arg1, arg2: direction
): vector;

LOCAL
    v2      : LIST [3:3] OF REAL;

```

```

    v1      : LIST [3:3] OF REAL;
    mag     : REAL;
    res     : direction;
    result  : vector;
END_LOCAL;
IF (NOT EXISTS(arg1)) OR (arg1.dim = 2) OR (NOT EXISTS(arg2)) OR (arg2
    .dim = 2) THEN
    RETURN(?);
ELSE
    BEGIN
        v1 := normalise(arg1).direction_ratios;
        v2 := normalise(arg2).direction_ratios;
        res.name := '';
        res.direction_ratios[1] := (v1[2] * v2[3]) - (v1[3] * v2[2]);
        res.direction_ratios[2] := (v1[3] * v2[1]) - (v1[1] * v2[3]);
        res.direction_ratios[3] := (v1[1] * v2[2]) - (v1[2] * v2[1]);
        mag := 0;
        REPEAT i := 1 TO 3 BY 1;
            mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
        END_REPEAT;
        IF mag > 0 THEN
            result.orientation := res;
            result.magnitude := SQRT(mag);
        ELSE
            result.orientation := arg1;
            result.magnitude := 0;
        END_IF;
        result.name := '';
        RETURN(result);
    END;
END_IF;

END_FUNCTION; -- cross_product

FUNCTION curve_weights_positive(
    b: rational_b_spline_curve
): BOOLEAN;

    LOCAL
        result : BOOLEAN := TRUE;
    END_LOCAL;
    REPEAT i := 0 TO b.upper_index_on_control_points BY 1;
        IF b.weights[i] <= 0 THEN
            result := FALSE;
            RETURN(result);
        END_IF;
    END_REPEAT;
    RETURN(result);

END_FUNCTION; -- curve_weights_positive

```

```

FUNCTION derive_dimensional_exponents(
    x: unit
): dimensional_exponents;

LOCAL
    i      : INTEGER;
    result : dimensional_exponents := dimensional_exponents(0,0,0,0,0,0,
        0);
END_LOCAL;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'DERIVED_UNIT'
    IN TYPEOF(x) THEN
    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements) BY 1;
        result.length_exponent := result.length_exponent + (x.elements[i].
            exponent * x.elements[i].unit.dimensions.length_exponent);
        result.mass_exponent := result.mass_exponent + (x.elements[i].
            exponent * x.elements[i].unit.dimensions.mass_exponent);
        result.time_exponent := result.time_exponent + (x.elements[i].
            exponent * x.elements[i].unit.dimensions.time_exponent);
        result.electric_current_exponent := result.
            electric_current_exponent + (x.elements[i].exponent * x.
            elements[i].unit.dimensions.electric_current_exponent);
        result.thermodynamic_temperature_exponent := result.
            thermodynamic_temperature_exponent + (x.elements[i].exponent *
            x.elements[i].unit.dimensions.
            thermodynamic_temperature_exponent);
        result.amount_of_substance_exponent := result.
            amount_of_substance_exponent + (x.elements[i].exponent * x.
            elements[i].unit.dimensions.amount_of_substance_exponent);
        result.luminous_intensity_exponent := result.
            luminous_intensity_exponent + (x.elements[i].exponent * x.
            elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
ELSE
    result := x.dimensions;
END_IF;
RETURN(result);

END_FUNCTION; -- derive_dimensional_exponents

```

```

FUNCTION dimension_of(
    item: geometric_representation_item
): dimension_count;

LOCAL
    x : SET OF representation;
    y : representation_context;
END_LOCAL;
x := using_representations(item);

```

```

y := x[1].context_of_items;
RETURN(y\geometric_representation_context.coordinate_space_dimension);

END_FUNCTION; -- dimension_of

FUNCTION dimensions_for_si_unit(
    n: si_unit_name
): dimensional_exponents;
CASE n OF
    metre      : RETURN(dimensional_exponents(1,0,0,0,0,0,0));
    gram       : RETURN(dimensional_exponents(0,1,0,0,0,0,0));
    second     : RETURN(dimensional_exponents(0,0,1,0,0,0,0));
    ampere     : RETURN(dimensional_exponents(0,0,0,1,0,0,0));
    kelvin     : RETURN(dimensional_exponents(0,0,0,0,1,0,0));
    mole       : RETURN(dimensional_exponents(0,0,0,0,0,1,0));
    candela    : RETURN(dimensional_exponents(0,0,0,0,0,0,1));
    radian     : RETURN(dimensional_exponents(0,0,0,0,0,0,0));
    steradian  : RETURN(dimensional_exponents(0,0,0,0,0,0,0));
    hertz      : RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
    newton     : RETURN(dimensional_exponents(1,1,-2,0,0,0,0));
    pascal     : RETURN(dimensional_exponents(-1,1,-2,0,0,0,0));
    joule      : RETURN(dimensional_exponents(2,1,-2,0,0,0,0));
    watt       : RETURN(dimensional_exponents(2,1,-3,0,0,0,0));
    coulomb    : RETURN(dimensional_exponents(0,0,1,1,0,0,0));
    volt       : RETURN(dimensional_exponents(2,1,-3,-1,0,0,0));
    farad      : RETURN(dimensional_exponents(-2,-1,4,1,0,0,0));
    ohm        : RETURN(dimensional_exponents(2,1,-3,-2,0,0,0));
    siemens    : RETURN(dimensional_exponents(-2,-1,3,2,0,0,0));
    weber      : RETURN(dimensional_exponents(2,1,-2,-1,0,0,0));
    tesla      : RETURN(dimensional_exponents(0,1,-2,-1,0,0,0));
    henry      : RETURN(dimensional_exponents(2,1,-2,-2,0,0,0));
    degree_celsius : RETURN(dimensional_exponents(0,0,0,0,1,0,0));
    lumen      : RETURN(dimensional_exponents(0,0,0,0,0,0,1));
    lux        : RETURN(dimensional_exponents(-2,0,0,0,0,0,1));
    becquerel  : RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
    gray       : RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
    sievert    : RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
END_CASE;

END_FUNCTION; -- dimensions_for_si_unit

FUNCTION dot_product(
    arg1, arg2: direction
): REAL;

LOCAL
    ndim : INTEGER;
    scalar : REAL;
    vec1 : direction;
    vec2 : direction;

```

```

END_LOCAL;
IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) THEN
    scalar := ?;
ELSE
    IF arg1.dim <> arg2.dim THEN
        scalar := ?;
    ELSE
        BEGIN
            vec1 := normalise(arg1);
            vec2 := normalise(arg2);
            ndim := arg1.dim;
            scalar := 0;
            REPEAT i := 1 TO ndim BY 1;
                scalar := scalar + (vec1.direction_ratios[i] * vec2.
                    direction_ratios[i]);
            END_REPEAT;
        END;
    END_IF;
END_IF;
RETURN(scalar);

END_FUNCTION; -- dot_product

FUNCTION edge_reversed(
    an_edge: edge
): oriented_edge;

LOCAL
    the_reverse : oriented_edge;
END_LOCAL;
the_reverse.name := '';
the_reverse.edge_start := an_edge.edge_end;
the_reverse.edge_end := an_edge.edge_start;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ORIENTED_EDGE'
    IN TYPEOF(an_edge) THEN
    the_reverse.edge_element := an_edge\oriented_edge.edge_element;
    the_reverse.orientation := NOT an_edge\oriented_edge.orientation;
ELSE
    the_reverse.edge_element := an_edge;
    the_reverse.orientation := FALSE;
END_IF;
RETURN(the_reverse);

END_FUNCTION; -- edge_reversed

FUNCTION face_bound_reversed(
    a_face_bound: face_bound
): face_bound;

```

```

LOCAL
  the_reverse : face_bound;
END_LOCAL;
the_reverse.name := '';
IF
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'FACE_OUTER_BOUND'
  IN TYPEOF(a_face_bound) THEN
    the_reverse.bound := a_face_bound\face_bound.bound;
    the_reverse.orientation := NOT a_face_bound\face_bound.orientation;
  ELSE
    the_reverse.bound := a_face_bound.bound;
    the_reverse.orientation := NOT a_face_bound.orientation;
  END_IF;
RETURN(the_reverse);

END_FUNCTION; -- face_bound_reversed

FUNCTION face_reversed(
  a_face: face
): oriented_face;

LOCAL
  the_reverse : oriented_face;
END_LOCAL;
the_reverse.name := '';
the_reverse.bound := set_of_topology_reversed(a_face.bound);
IF
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'ORIENTED_FACE'
  IN TYPEOF(a_face) THEN
    the_reverse.face_element := a_face\oriented_face.face_element;
    the_reverse.orientation := NOT a_face\oriented_face.orientation;
  ELSE
    the_reverse.face_element := a_face;
    the_reverse.orientation := FALSE;
  END_IF;
RETURN(the_reverse);

END_FUNCTION; -- face_reversed

FUNCTION first_proj_axis(
  z_axis, arg: direction
): direction;

LOCAL
  x_vec : vector;
  v      : direction;
  z      : direction;

```

```

    x_axis : direction;
END_LOCAL;
IF NOT EXISTS(z_axis) THEN
    RETURN(?);
ELSE
    z := normalise(z_axis);
    IF NOT EXISTS(arg) THEN
        IF z.direction_ratios <> [1,0,0] THEN
            v := dummy_gri || direction([1,0,0]);
        ELSE
            v := dummy_gri || direction([0,1,0]);
        END_IF;
    ELSE
        IF arg.dim <> 3 THEN
            RETURN(?);
        END_IF;
        IF cross_product(arg,z).magnitude = 0 THEN
            RETURN(?);
        ELSE
            v := normalise(arg);
        END_IF;
    END_IF;
    x_vec := scalar_times_vector(dot_product(v,z),z);
    x_axis := vector_difference(v,x_vec).orientation;
    x_axis := normalise(x_axis);
END_IF;
RETURN(x_axis);

END_FUNCTION; -- first_proj_axis

FUNCTION gbsf_check_curve(
    cv: curve
): BOOLEAN;
IF SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOUNDED_CURVE',

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CIRCLE',

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELLIPSE'] *
    TYPEOF(cv)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'B_SPLINE_CURVE'
        IN TYPEOF(cv)) AND (cv\b_spline_curve.self_intersect = FALSE)
    THEN
        RETURN(TRUE);
    END_IF;
END_IF;
END_FUNCTION;

```



```

ELSE
  IF (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'COMPOSITE_CURVE'
    IN TYPEOF(cv)) AND (cv\composite_curve.self_intersect = FALSE)
    THEN
      RETURN(SIZEOF(QUERY ( seg <* cv\composite_curve.segments | (NOT
        gbsf_check_curve(seg.parent_curve)) )) = 0);
    ELSE
      IF SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'BOUNDED_CURVE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CURVE_REPLICA'])
        * TYPEOF(cv)) = 1 THEN
        RETURN(gbsf_check_curve(cv\curve_replica.parent_curve));
      ELSE
        IF (SIZEOF([
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'BOUNDED_CURVE',
          'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
          'OFFSET_CURVE_3D'])
          * TYPEOF(cv)) = 1) AND (cv\offset_curve_3d.self_intersect =
            FALSE) THEN
            RETURN(gbsf_check_curve(cv\offset_curve_3d.basis_curve));
          ELSE
            IF SIZEOF([
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'BOUNDED_CURVE',
              'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
              'PCURVE'])
              * TYPEOF(cv)) = 1 THEN
              RETURN(gbsf_check_curve(cv\pcurve.reference_to_curve.items
                [1]) AND gbsf_check_surface(cv\pcurve.basis_surface));
            ELSE
              IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'POLYLINE'
                IN TYPEOF(cv) THEN
                IF (SIZEOF(cv\polyline.points) >= 3) AND (SIZEOF(
                  bag_to_set(USEDIN(cv, '')) - bag_to_set(USEDIN(cv,
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'SURFACE_CURVE.CURVE_3D'))))
                  = 0) THEN
                  RETURN(TRUE);
                END_IF;
              ELSE
                IF SIZEOF([
                  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                  'BOUNDED_CURVE',
                  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                  'SURFACE_CURVE'])

```

```

        * TYPEOF(cv)) = 1 THEN
    IF gbsf_check_curve(cv\surface_curve.curve_3d) THEN
        REPEAT i := 1 TO SIZEOF(cv\surface_curve.
            associated_geometry) BY 1;
            IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE'
                IN TYPEOF(cv\surface_curve.associated_geometry[
                    i]) THEN
                    IF NOT gbsf_check_surface(cv\surface_curve.
                        associated_geometry[i]) THEN
                        RETURN(FALSE);
                    END_IF;
                ELSE
                    IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PCURVE'
                        IN TYPEOF(cv\surface_curve.
                            associated_geometry[i]) THEN
                            IF NOT gbsf_check_curve(cv\surface_curve.
                                associated_geometry[i]) THEN
                                RETURN(FALSE);
                            END_IF;
                        END_IF;
                    END_IF;
                END_REPEAT;
                RETURN(TRUE);
            END_IF;
        ELSE
            IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'TRIMMED_CURVE'
                IN TYPEOF(cv) THEN
                IF SIZEOF([
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'BOUNDED_CURVE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINE',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PARABOLA',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'HYPERBOLA'])
                    * TYPEOF(cv\trimmed_curve.basis_curve)) = 1 THEN
                    RETURN(TRUE);
                ELSE
                    RETURN(gbsf_check_curve(cv\trimmed_curve.
                        basis_curve));
                END_IF;
            END_IF;
        END_IF;
    END_IF;
END_IF;

```

```

        END_IF;
    END_IF;
END_IF;
END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- gbsf_check_curve

FUNCTION gbsf_check_point(
    pnt: point
): BOOLEAN;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_POINT'
    IN TYPEOF(pnt) THEN
    RETURN(TRUE);
ELSE
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POINT_ON_CURVE'
        IN TYPEOF(pnt) THEN
        RETURN(gbsf_check_curve(pnt\point_on_curve.basis_curve));
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'POINT_ON_SURFACE'
            IN TYPEOF(pnt) THEN
            RETURN(gbsf_check_surface(pnt\point_on_surface.basis_surface));
        ELSE
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'DEGENERATE_PCURVE'
                IN TYPEOF(pnt) THEN
                RETURN(gbsf_check_curve(pnt\degenerate_pcurve.
                    reference_to_curve.items[1]) AND gbsf_check_surface(pnt\
                    degenerate_pcurve.basis_surface));
            END_IF;
        END_IF;
    END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- gbsf_check_point

FUNCTION gbsf_check_surface(
    sf: surface
): BOOLEAN;
IF (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'B_SPLINE_SURFACE'
    IN TYPEOF(sf)) AND (sf\b_spline_surface.self_intersect = FALSE)
    THEN
    RETURN(TRUE);
ELSE
    IF SIZEOF([
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'SPHERICAL_SURFACE',
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TOROIDAL_SURFACE']
        * TYPEOF(sf)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CURVE_BOUNDED_SURFACE'
        IN TYPEOF(sf) THEN
        IF SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CONICAL_SURFACE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CYLINDRICAL_SURFACE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PLANE']
            * TYPEOF(sf\curve_bounded_surface.basis_surface)) = 1 THEN
            RETURN(SIZEOF(QUERY ( bcurve <* sf\curve_bounded_surface.
                boundaries | (NOT gbsf_check_curve(bcurve)) )) = 0);
        ELSE
            IF gbsf_check_surface(sf\curve_bounded_surface.basis_surface)
            THEN
                RETURN(SIZEOF(QUERY ( bcurve <* sf\curve_bounded_surface.
                    boundaries | (NOT gbsf_check_curve(bcurve)) )) = 0);
            END_IF;
        END_IF;
    ELSE
        IF (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'OFFSET_SURFACE'
            IN TYPEOF(sf)) AND (sf\offset_surface.self_intersect = FALSE)
            THEN
                RETURN(gbsf_check_surface(sf\offset_surface.basis_surface));
        ELSE
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'RECTANGULAR_COMPOSITE_SURFACE'
                IN TYPEOF(sf) THEN
                REPEAT i := 1 TO SIZEOF(sf\rectangular_composite_surface.
                    segments) BY 1;
                REPEAT j := 1 TO SIZEOF(sf\rectangular_composite_surface.
                    segments[i]) BY 1;
                IF NOT gbsf_check_surface(sf\
                    rectangular_composite_surface.segments[i][j].

```

```

        parent_surface) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_REPEAT;
RETURN(TRUE);
ELSE
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RECTANGULAR_TRIMMED_SURFACE'
        IN TYPEOF(sf) THEN
            IF SIZEOF([
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CONICAL_SURFACE',
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CYLINDRICAL_SURFACE',
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'PLANE']
                * TYPEOF(sf\rectangular_trimmed_surface.basis_surface))
                = 1 THEN
                RETURN(TRUE);
            ELSE
                RETURN(gbsf_check_surface(sf\rectangular_trimmed_surface
                    .basis_surface));
            END_IF;
        ELSE
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'SURFACE_REPLICA'
                IN TYPEOF(sf) THEN
                RETURN(gbsf_check_surface(sf\surface_replica.
                    parent_surface));
            ELSE
                IF
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'SWEPT_SURFACE'
                    IN TYPEOF(sf) THEN
                    RETURN(gbsf_check_curve(sf\swept_surface.swept_curve));
                END_IF;
            END_IF;
        END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    RETURN(FALSE);
END_FUNCTION; -- gbsf_check_surface

FUNCTION get_basis_surface(

```

```

        c: curve_on_surface
    ): SET [0:2] OF surface;

LOCAL
    surfs : SET [0:2] OF surface;
    n      : INTEGER;
END_LOCAL;
surfs := [];
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PCURVE' IN
    TYPEOF(c) THEN
        surfs := [c\pcurve.basis_surface];
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SURFACE_CURVE'
            IN TYPEOF(c) THEN
                n := SIZEOF(c\surface_curve.associated_geometry);
                REPEAT i := 1 TO n BY 1;
                    surfs := surfs + associated_surface(c\surface_curve.
                        associated_geometry[i]);
                END_REPEAT;
            END_IF;
        END_IF;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'COMPOSITE_CURVE_ON_SURFACE'
        IN TYPEOF(c) THEN
            n := SIZEOF(c\composite_curve.segments);
            surfs := get_basis_surface(c\composite_curve.segments[1].
                parent_curve);
            IF n > 1 THEN
                REPEAT i := 2 TO n BY 1;
                    surfs := surfs * get_basis_surface(c\composite_curve.segments[i]
                        .parent_curve);
                END_REPEAT;
            END_IF;
        END_IF;
    RETURN(surfs);

END_FUNCTION; -- get_basis_surface

FUNCTION item_in_context(
    item: representation_item;
    cntxt: representation_context
): BOOLEAN;

LOCAL
    i : INTEGER;
    y : BAG OF representation_item;

```

```

END_LOCAL;
IF SIZEOF(USEDIN(item,
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'REPRESENTATION.ITEMS')
  * cntxt.representations_in_context) > 0 THEN
  RETURN(TRUE);
ELSE
  y := QUERY ( z <* USEDIN(item, '') | (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_ITEM'
    IN TYPEOF(z)) );
  IF SIZEOF(y) > 0 THEN
    REPEAT i := 1 TO HIINDEX(y) BY 1;
      IF item_in_context(y[i], cntxt) THEN
        RETURN(TRUE);
      END_IF;
    END_REPEAT;
  END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- item_in_context

FUNCTION leap_year(
  year: year_number
): BOOLEAN;
IF ((year MOD 4) = 0) AND ((year MOD 100) <> 0) OR ((year MOD 400) =
  0) THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;

END_FUNCTION; -- leap_year

FUNCTION list_face_loops(
  f: face
): LIST [0:?] OF loop;

LOCAL
  loops : LIST [0:?] OF loop := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(f.bounds) BY 1;
  loops := loops + f.bounds[i].bound;
END_REPEAT;
RETURN(loops);

END_FUNCTION; -- list_face_loops

```

```

FUNCTION list_of_topology_reversed(
    a_list: list_of_reversible_topology_item
): list_of_reversible_topology_item;

LOCAL
    the_reverse : list_of_reversible_topology_item;
END_LOCAL;
the_reverse := [];
REPEAT i := 1 TO SIZEOF(a_list) BY 1;
    the_reverse := topology_reversed(a_list[i]) + the_reverse;
END_REPEAT;
RETURN(the_reverse);

END_FUNCTION; -- list_of_topology_reversed

```

```

FUNCTION list_to_array(
    lis: LIST [0:?] OF GENERIC:t;
    low, u: INTEGER
): ARRAY [low:u] OF GENERIC:t;

LOCAL
    n : INTEGER;
    res : ARRAY [low:u] OF GENERIC:t;
END_LOCAL;
n := SIZEOF(lis);
IF n <> ((u - low) + 1) THEN
    RETURN(?);
ELSE
    REPEAT i := 1 TO n BY 1;
        res[(low + i) - 1] := lis[i];
    END_REPEAT;
    RETURN(res);
END_IF;

END_FUNCTION; -- list_to_array

```

```

FUNCTION list_to_set(
    l: LIST [0:?] OF GENERIC:t
): SET OF GENERIC:t;

LOCAL
    s : SET OF GENERIC:t := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(l) BY 1;
    s := s + l[i];
END_REPEAT;
RETURN(s);

END_FUNCTION; -- list_to_set

```



```

FUNCTION make_array_of_array(
    lis: LIST [1:?] OF LIST [1:?] OF GENERIC:t;
    low1, u1, low2, u2: INTEGER
): ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;

LOCAL
    n2 : INTEGER;
    n1 : INTEGER;
    res : ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;
    resl : LIST [1:?] OF ARRAY [low2:u2] OF GENERIC:t;
END_LOCAL;
n1 := SIZEOF(lis);
n2 := SIZEOF(lis[1]);
IF (n1 <> ((u1 - low1) + 1)) AND (n2 <> ((u2 - low2) + 1)) THEN
    RETURN(?);
END_IF;
REPEAT i := 1 TO n1 BY 1;
    IF SIZEOF(lis[i]) <> n2 THEN
        RETURN(?);
    END_IF;
END_REPEAT;
REPEAT i := 1 TO n1 BY 1;
    resl[i] := list_to_array(lis[i], low2, u2);
END_REPEAT;
res := list_to_array(resl, low1, u1);
RETURN(res);

END_FUNCTION; -- make_array_of_array

```

```

FUNCTION mixed_loop_type_set(
    l: SET [0:?] OF loop
): LOGICAL;

LOCAL
    i : INTEGER;
    poly_loop_type : LOGICAL;
END_LOCAL;
IF SIZEOF(l) <= 1 THEN
    RETURN(FALSE);
END_IF;
poly_loop_type :=

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POLY_LOOP'
    IN TYPEOF(l[1]);
REPEAT i := 2 TO SIZEOF(l) BY 1;
    IF (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POLY_LOOP'
        IN TYPEOF(l[i])) <> poly_loop_type THEN

```

```

        RETURN(TRUE);
    END_IF;
END_REPEAT;
RETURN(FALSE);

END_FUNCTION; -- mixed_loop_type_set

FUNCTION msb_shells(
    brep: manifold_solid_brep;
    schema_name: STRING
): SET [1:?] OF closed_shell;
IF (schema_name + '.BREP_WITH_VOIDS') IN TYPEOF(brep) THEN
    RETURN(brep\brep_with_voids.voids + brep.outer);
ELSE
    RETURN([brep.outer]);
END_IF;

END_FUNCTION; -- msb_shells

FUNCTION msf_curve_check(
    cv: curve
): BOOLEAN;
IF SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'BOUNDED_CURVE',

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CONIC',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE_REPLICA',

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LINE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'OFFSET_CURVE_3D']
    * TYPEOF(cv)) > 1 THEN
    RETURN(FALSE);
ELSE
    IF (
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'B_SPLINE_CURVE'
        IN TYPEOF(cv)) AND (cv\b_spline_curve.self_intersect = FALSE)
    THEN
        RETURN(TRUE);
    ELSE
        IF SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'CONIC',

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINE'] *
    TYPEOF(cv)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CURVE_REPLICA'
        IN TYPEOF(cv) THEN
        RETURN(msf_curve_check(cv\curve_replica.parent_curve));
    ELSE
        IF (
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'OFFSET_CURVE_3D'
            IN TYPEOF(cv)) AND (cv\offset_curve_3d.self_intersect =
            FALSE) THEN
            RETURN(msf_curve_check(cv\offset_curve_3d.basis_curve));
        ELSE
            IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PCURVE'
                IN TYPEOF(cv) THEN
                RETURN(msf_curve_check(cv\pcurve.reference_to_curve\
                representation.items[1]) AND msf_surface_check(cv\
                pcurve.basis_surface));
            ELSE
                IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE_CURVE'
                    IN TYPEOF(cv) THEN
                    IF msf_curve_check(cv\surface_curve.curve_3d) THEN
                        REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                        associated_geometry) BY 1;
                        IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'SURFACE'
                            IN TYPEOF(cv\surface_curve.associated_geometry[i])
                            THEN
                            IF NOT msf_surface_check(cv\surface_curve.
                            associated_geometry[i]) THEN
                                RETURN(FALSE);
                            END_IF;
                        ELSE
                            IF
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'PCURVE'
                                IN TYPEOF(cv\surface_curve.associated_geometry[
                                i]) THEN
                                IF NOT msf_curve_check(cv\surface_curve.
                                associated_geometry[i]) THEN
                                    RETURN(FALSE);
                                END_IF;

```

```

        END_IF;
        END_IF;
        END_REPEAT;
        RETURN(TRUE);
    END_IF;
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'POLYLINE'
                IN TYPEOF(cv) THEN
                IF SIZEOF(cv\polyline.points) >= 3 THEN
                    RETURN(TRUE);
                END_IF;
            END_IF;
        END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    RETURN(FALSE);
END_FUNCTION; -- msf_curve_check

FUNCTION msf_surface_check(
    surf: surface
): BOOLEAN;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELEMENTARY_SURFACE'
        IN TYPEOF(surf) THEN
        RETURN(TRUE);
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'SWEPT_SURFACE'
                IN TYPEOF(surf) THEN
                RETURN(msf_curve_check(surf\swept_surface.swept_curve));
            ELSE
                IF (
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'OFFSET_SURFACE'
                        IN TYPEOF(surf)) AND (surf\offset_surface.self_intersect =
                    FALSE) THEN
                    RETURN(msf_surface_check(surf\offset_surface.basis_surface));
                ELSE
                    IF
                        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                        'SURFACE_REPLICA'
                            IN TYPEOF(surf) THEN

```

```

        RETURN(msf_surface_check(surf\surface_replica.parent_surface));
    ELSE
        IF (
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'B_SPLINE_SURFACE'
                IN TYPEOF(surf)) AND (surf\b_spline_surface.self_intersect
                = FALSE) THEN
            RETURN(TRUE);
        END_IF;
    END_IF;
END_IF;
END_IF;
RETURN(FALSE);
END_FUNCTION; -- msf_surface_check

```

```

FUNCTION normalise(
    arg: vector_or_direction
): vector_or_direction;

LOCAL
    ndim    : INTEGER;
    v        : direction;
    vec      : vector;
    mag      : REAL;
    result   : vector_or_direction;
END_LOCAL;
IF NOT EXISTS(arg) THEN
    result := ?;
ELSE
    ndim := arg.dim;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VECTOR' IN
            TYPEOF(arg) THEN
        BEGIN
            vec := arg;
            v := arg.orientation;
            IF arg.magnitude = 0 THEN
                RETURN(?);
            ELSE
                vec.magnitude := 1;
            END_IF;
        END;
    ELSE
        v := arg;
    END_IF;
    mag := 0;
    REPEAT i := 1 TO ndim BY 1;
        mag := mag + (v.direction_ratios[i] * v.direction_ratios[i]);
    END_REPEAT;
END_IF;

```

```

END_REPEAT;
IF mag > 0 THEN
    mag := SQRT(mag);
    REPEAT i := 1 TO ndim BY 1;
        v.direction_ratios[i] := v.direction_ratios[i] / mag;
    END_REPEAT;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VECTOR'
        IN TYPEOF(arg) THEN
        vec.orientation := v;
        result := vec;
    ELSE
        result := v;
    END_IF;
ELSE
    RETURN(?);
END_IF;
END_IF;
RETURN(result);

END_FUNCTION; -- normalise

FUNCTION orthogonal_complement(
    vec: direction
): direction;

LOCAL
    result : direction;
END_LOCAL;
IF (vec.dim <> 2) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
ELSE
    result.name := '';
    result.direction_ratios[1] := -vec.direction_ratios[2];
    result.direction_ratios[2] := vec.direction_ratios[1];
    RETURN(result);
END_IF;

END_FUNCTION; -- orthogonal_complement

FUNCTION path_head_to_tail(
    a_path: path
): LOGICAL;

LOCAL
    n : INTEGER;
    p : BOOLEAN := TRUE;
END_LOCAL;
n := SIZEOF(a_path.edge_list);

```

```

REPEAT i := 2 TO n BY 1;
  p := p AND (a_path.edge_list[i - 1].edge_end :=: a_path.edge_list[i]
    .edge_start);
END_REPEAT;
RETURN(p);

END_FUNCTION; -- path_head_to_tail


FUNCTION path_reversed(
  a_path: path
): oriented_path;

LOCAL
  the_reverse : oriented_path;
END_LOCAL;
the_reverse.name := '';
the_reverse.edge_list := list_of_topology_reversed(a_path.edge_list);
IF
  'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
  'ORIENTED_PATH'
  IN TYPEOF(a_path) THEN
  the_reverse.path_element := a_path\oriented_path.path_element;
  the_reverse.orientation := NOT a_path\oriented_path.orientation;
ELSE
  the_reverse.path_element := a_path;
  the_reverse.orientation := FALSE;
END_IF;
RETURN(the_reverse);

END_FUNCTION; -- path_reversed


FUNCTION scalar_times_vector(
  scalar: REAL;
  vec: vector_or_direction
): vector;

LOCAL
  v      : direction;
  mag    : REAL;
  result : vector;
END_LOCAL;
IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
  RETURN(?);
ELSE
  IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'VECTOR' IN
    TYPEOF(vec) THEN
    v := vec.orientation;
    mag := scalar * vec.magnitude;

```

```

ELSE
    v := vec;
    mag := scalar;
END_IF;
IF mag < 0 THEN
    REPEAT i := 1 TO SIZEOF(v.direction_ratios) BY 1;
        v.direction_ratios[i] := -v.direction_ratios[i];
    END_REPEAT;
    mag := -mag;
END_IF;
result.name := '';
result.orientation := normalise(v);
result.magnitude := mag;
END_IF;
RETURN(result);

END_FUNCTION; -- scalar_times_vector

FUNCTION second_proj_axis(
    z_axis, x_axis, arg: direction
): direction;

LOCAL
    temp : vector;
    v : direction;
    y_axis : vector;
END_LOCAL;
IF NOT EXISTS(arg) THEN
    v := dummy_gri || direction([0,1,0]);
ELSE
    v := arg;
END_IF;
temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
y_axis := vector_difference(v,temp);
temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
y_axis := vector_difference(y_axis,temp);
y_axis := normalise(y_axis);
RETURN(y_axis.orientation);

END_FUNCTION; -- second_proj_axis

FUNCTION set_of_topology_reversed(
    a_set: set_of_reversible_topology_item
): set_of_reversible_topology_item;

LOCAL
    the_reverse : set_of_reversible_topology_item;
END_LOCAL;
the_reverse := [];
REPEAT i := 1 TO SIZEOF(a_set) BY 1;

```



```

        the_reverse := the_reverse + topology_reversed(a_set[i]);
    END_REPEAT;
    RETURN(the_reverse);

END_FUNCTION; -- set_of_topology_reversed

FUNCTION shell_reversed(
    a_shell: shell
): shell;

LOCAL
    the_reverse : shell;
END_LOCAL;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ORIENTED_OPEN_SHELL'
    IN TYPEOF(a_shell) THEN
    the_reverse := representation_item('') ||
        topological_representation_item() || connected_face_set(
            set_of_topology_reversed(a_shell\connected_face_set.cfs_faces))
        || open_shell() || oriented_open_shell(a_shell\
            oriented_open_shell.open_shell_element, NOT a_shell\
            oriented_open_shell.orientation);
ELSE
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'OPEN_SHELL'
        IN TYPEOF(a_shell) THEN
        the_reverse := representation_item('') ||
            topological_representation_item() || connected_face_set(
                set_of_topology_reversed(a_shell\connected_face_set.cfs_faces))
                || open_shell() || oriented_open_shell(a_shell, FALSE);
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'ORIENTED_CLOSED_SHELL'
            IN TYPEOF(a_shell) THEN
            the_reverse := representation_item('') ||
                topological_representation_item() || connected_face_set(
                    set_of_topology_reversed(a_shell\connected_face_set.cfs_faces))
                    || closed_shell() || oriented_closed_shell(a_shell\
                        oriented_closed_shell.closed_shell_element, NOT a_shell\
                        oriented_closed_shell.orientation);
        ELSE
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CLOSED_SHELL'
                IN TYPEOF(a_shell) THEN
                the_reverse := representation_item('') ||
                    topological_representation_item() || connected_face_set(
                        set_of_topology_reversed(a_shell\connected_face_set.

```

```

        cfs_faces)) || closed_shell() || oriented_closed_shell(
        a_shell,FALSE);
    ELSE
        RETURN(?);
    END_IF;
END_IF;
END_IF;
END_IF;
RETURN(the_reverse);

END_FUNCTION; -- shell_reversed

FUNCTION surface_weights_positive(
    b: rational_b_spline_surface
): BOOLEAN;

LOCAL
    result : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 0 TO b.u_upper BY 1;
    REPEAT j := 0 TO b.v_upper BY 1;
        IF b.weights[i][j] <= 0 THEN
            result := FALSE;
            RETURN(result);
        END_IF;
    END_REPEAT;
END_REPEAT;
RETURN(result);

END_FUNCTION; -- surface_weights_positive

FUNCTION topology_reversed(
    an_item: reversible_topology
): reversible_topology;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'EDGE' IN
    TYPEOF(an_item) THEN
        RETURN(edge_reversed(an_item));
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PATH' IN
    TYPEOF(an_item) THEN
        RETURN(path_reversed(an_item));
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACE_BOUND'
    IN TYPEOF(an_item) THEN

```

```

    RETURN(face_bound_reversed(an_item));
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FACE' IN
    TYPEOF(an_item) THEN
    RETURN(face_reversed(an_item));
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SHELL' IN
    TYPEOF(an_item) THEN
    RETURN(shell_reversed(an_item));
END_IF;
IF 'SET' IN TYPEOF(an_item) THEN
    RETURN(set_of_topology_reversed(an_item));
END_IF;
IF 'LIST' IN TYPEOF(an_item) THEN
    RETURN(list_of_topology_reversed(an_item));
END_IF;
RETURN(?);

END_FUNCTION; -- topology_reversed

FUNCTION unique_version_change_order(
    c: action;
    schema_name: STRING
): BOOLEAN;

LOCAL
    ords      : action_directive := c\directed_action.directive;
    assign    : SET OF action_request_assignment := [];
    versions  : SET OF product_definition_formation := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(ords.requests) BY 1;
    assign := assign + QUERY ( ara <* bag_to_set(USEDIN(ords.requests[i],
        schema_name + '.ACTION_REQUEST_ASSIGNMENT.' +
        'ASSIGNED_ACTION_REQUEST')) | ((schema_name + '.CHANGE_REQUEST')
        IN TYPEOF(ara)) );
END_REPEAT;
REPEAT k := 1 TO SIZEOF(assign) BY 1;
    versions := versions + assign[k]\change_request.items;
END_REPEAT;
RETURN(SIZEOF(QUERY ( vers <* versions | (NOT (SIZEOF(
    QUERY ( other_vers <* (versions - vers) | (vers.of_product :=:
    other_vers.of_product) )) = 0)) )) = 0);

END_FUNCTION; -- unique_version_change_order

FUNCTION using_items(

```

```

        item: founded_item_select;
        checked_items: SET OF founded_item_select
    ): SET OF founded_item_select;

LOCAL
    next_items      : SET OF founded_item_select;
    new_check_items : SET OF founded_item_select;
    result_items    : SET OF founded_item_select;
END_LOCAL;
result_items := [];
new_check_items := checked_items + item;
next_items := QUERY ( z <* bag_to_set(USEDIN(item, '')) | ((
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION_ITEM'
    IN TYPEOF(z)) OR (
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'FOUNDED_ITEM'
    IN TYPEOF(z))) );
IF SIZEOF(next_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(next_items) BY 1;
        IF NOT (next_items[i] IN new_check_items) THEN
            result_items := result_items + next_items[i] + using_items(
                next_items[i], new_check_items);
        END_IF;
    END_REPEAT;
END_IF;
RETURN(result_items);

END_FUNCTION; -- using_items

FUNCTION using_representations(
    item: founded_item_select
): SET OF representation;

LOCAL
    results      : SET OF representation;
    intermediate_items : SET OF founded_item_select;
    result_bag    : BAG OF representation;
END_LOCAL;
results := [];
result_bag := USEDIN(item,
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'REPRESENTATION.ITEMS');
IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag) BY 1;
        results := results + result_bag[i];
    END_REPEAT;
END_IF;
intermediate_items := using_items(item, []);
IF SIZEOF(intermediate_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(intermediate_items) BY 1;

```

```

        result_bag := USEDIN(intermediate_items[i],
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'REPRESENTATION.ITEMS');
    IF SIZEOF(result_bag) > 0 THEN
        REPEAT j := 1 TO HIINDEX(result_bag) BY 1;
            results := results + result_bag[j];
        END_REPEAT;
    END_IF;
END_REPEAT;
END_IF;
RETURN(results);

END_FUNCTION; -- using_representations

FUNCTION valid_2d_wireframe_edge_curve(
    crv: curve;
    schma: STRING
): BOOLEAN;
IF SIZEOF([schma + '.LINE', schma + '.B_SPLINE_CURVE', schma + '.CIRCLE',
    schma + '.HYPERBOLA', schma + '.ELLIPSE', schma + '.PARABOLA', schma +
    '.POLYLINE'] * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF (schma + '.CURVE_REPLICA') IN TYPEOF(crv) THEN
        RETURN(valid_2d_wireframe_edge_curve(crv\curve_replica.
            parent_curve, schma));
    ELSE
        IF (schma + '.OFFSET_CURVE_2D') IN TYPEOF(crv) THEN
            RETURN(valid_2d_wireframe_edge_curve(crv\offset_curve_2d.
                basis_curve, schma));
        END_IF;
    END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- valid_2d_wireframe_edge_curve

FUNCTION valid_basis_curve_in_2d_wireframe(
    crv: curve
): BOOLEAN;
IF SIZEOF([
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'POLYLINE',
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'B_SPLINE_CURVE',

    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELLIPSE',

```

```

        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'CIRCLE'] *
        TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'TRIMMED_CURVE'
        IN TYPEOF(crv) THEN
        IF SIZEOF([
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'LINE',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'PARABOLA',
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'HYPERBOLA']
            * TYPEOF(crv\trimmed_curve.basis_curve)) = 1 THEN
            RETURN(TRUE);
        ELSE
            RETURN(valid_basis_curve_in_2d_wireframe(crv\trimmed_curve.
                basis_curve));
        END_IF;
    ELSE
        IF
            'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
            'OFFSET_CURVE_2D'
            IN TYPEOF(crv) THEN
            RETURN(valid_basis_curve_in_2d_wireframe(crv\offset_curve_2d.
                basis_curve));
        ELSE
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'CURVE_REPLICA'
                IN TYPEOF(crv) THEN
                RETURN(valid_basis_curve_in_2d_wireframe(crv\curve_replica.
                    parent_curve));
            ELSE
                IF
                    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                    'COMPOSITE_CURVE'
                    IN TYPEOF(crv) THEN
                    RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments |
                        (NOT valid_basis_curve_in_2d_wireframe(ccs.parent_curve)) ))
                        = 0);
                END_IF;
            END_IF;
        END_IF;
    END_IF;
END_IF;
RETURN(FALSE);

```

```
END_FUNCTION; -- valid_basis_curve_in_2d_wireframe
```

```
FUNCTION valid_calendar_date(
    date: calendar_date
): LOGICAL;
IF NOT ((1 <= date.day_component) AND (date.day_component <= 31))
THEN
    RETURN(FALSE);
END_IF;
CASE date.month_component OF
    4      : RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
    6      : RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
    9      : RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
    11     : RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
    2      : BEGIN
        IF leap_year(date.year_component) THEN
            RETURN((1 <= date.day_component) AND (date.day_component <= 29));
        ELSE
            RETURN((1 <= date.day_component) AND (date.day_component <= 28));
        END_IF;
    END;
    OTHERWISE : RETURN(TRUE);
END_CASE;
```

```
END_FUNCTION; -- valid_calendar_date
```

```
FUNCTION valid_geometrically_bounded_wf_curve(
    crv: curve;
    schma: STRING
): BOOLEAN;
IF SIZEOF([schma + '.POLYLINE', schma + '.B_SPLINE_CURVE', schma +
    '.ELLIPSE', schma + '.CIRCLE'] * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
ELSE
    IF (schma + '.TRIMMED_CURVE') IN TYPEOF(crv) THEN
        IF SIZEOF([schma + '.LINE', schma + '.PARABOLA', schma +
            '.HYPERBOLA'] * TYPEOF(crv\trimmed_curve.basis_curve)) = 1
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(valid_geometrically_bounded_wf_curve(crv\trimmed_curve.
                basis_curve, schma));
        END_IF;
    ELSE
        IF (schma + '.OFFSET_CURVE_3D') IN TYPEOF(crv) THEN
            RETURN(valid_geometrically_bounded_wf_curve(crv\offset_curve_3d.
```

```

        basis_curve, schma));
ELSE
    IF (schma + '.CURVE_REPLICA') IN TYPEOF(crv) THEN
        RETURN(valid_geometrically_bounded_wf_curve(crv\curve_replica.
            parent_curve, schma));
    ELSE
        IF (schma + '.COMPOSITE_CURVE') IN TYPEOF(crv) THEN
            RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments |
                (NOT valid_geometrically_bounded_wf_curve(ccs.
                    parent_curve, schma)) )) = 0);
        END_IF;
    END_IF;
END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_curve

FUNCTION valid_geometrically_bounded_wf_point(
    pnt: point;
    schma: STRING
): BOOLEAN;
IF (schma + '.CARTESIAN_POINT') IN TYPEOF(pnt) THEN
    RETURN(TRUE);
ELSE
    IF (schma + '.POINT_ON_CURVE') IN TYPEOF(pnt) THEN
        RETURN(valid_geometrically_bounded_wf_curve(pnt\point_on_curve.
            basis_curve, schma));
    ELSE
        IF (schma + '.POINT_REPLICA') IN TYPEOF(pnt) THEN
            RETURN(valid_geometrically_bounded_wf_point(pnt\point_replica.
                parent_pt, schma));
        END_IF;
    END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_point

FUNCTION valid_measure_value(
    m: measure_value
): BOOLEAN;
IF 'REAL' IN TYPEOF(m) THEN
    RETURN(m > 0);
ELSE
    IF 'INTEGER' IN TYPEOF(m) THEN
        RETURN(m > 0);
    ELSE
        RETURN(TRUE);
    END_IF;
END_IF;

```



```

        END_IF;
    END_IF;

END_FUNCTION; -- valid_measure_value


FUNCTION valid_time(
    time: local_time
): BOOLEAN;
IF EXISTS(time.second_component) THEN
    RETURN(EXISTS(time.minute_component));
ELSE
    RETURN(TRUE);
END_IF;

END_FUNCTION; -- valid_time


FUNCTION valid_units(
    m: measure_with_unit
): BOOLEAN;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LENGTH_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'MASS_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,1,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'TIME_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,1,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'ELECTRIC_CURRENT_MEASURE'
    IN TYPEOF(m.value_component) THEN

```

```

    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,1,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'THERMODYNAMIC_TEMPERATURE_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,1,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'AMOUNT_OF_SUBSTANCE_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,1,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'LUMINOUS_INTENSITY_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,1) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'PLANE_ANGLE_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'SOLID_ANGLE_MEASURE'
    IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +

```

```

        'AREA_MEASURE'
        IN TYPEOF(m.value_component) THEN
        IF derive_dimensional_exponents(m.unit_component) <>
            dimensional_exponents(2,0,0,0,0,0,0) THEN
            RETURN(FALSE);
        END_IF;
    END_IF;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'VOLUME_MEASURE'
        IN TYPEOF(m.value_component) THEN
        IF derive_dimensional_exponents(m.unit_component) <>
            dimensional_exponents(3,0,0,0,0,0,0) THEN
            RETURN(FALSE);
        END_IF;
    END_IF;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'RATIO_MEASURE'
        IN TYPEOF(m.value_component) THEN
        IF derive_dimensional_exponents(m.unit_component) <>
            dimensional_exponents(0,0,0,0,0,0,0) THEN
            RETURN(FALSE);
        END_IF;
    END_IF;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITIVE_LENGTH_MEASURE'
        IN TYPEOF(m.value_component) THEN
        IF derive_dimensional_exponents(m.unit_component) <>
            dimensional_exponents(1,0,0,0,0,0,0) THEN
            RETURN(FALSE);
        END_IF;
    END_IF;
    IF
        'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
        'POSITIVE_PLANE_ANGLE_MEASURE'
        IN TYPEOF(m.value_component) THEN
        IF derive_dimensional_exponents(m.unit_component) <>
            dimensional_exponents(0,0,0,0,0,0,0) THEN
            RETURN(FALSE);
        END_IF;
    END_IF;
    RETURN(TRUE);

END_FUNCTION; -- valid_units

FUNCTION valid_wireframe_edge_curve(
    crv: curve
): BOOLEAN;
IF SIZEOF([

```

```

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'LINE',

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'CONIC',
'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'B_SPLINE_CURVE',

'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
'POLYLINE'] *
  TYPEOF(crv)) = 1 THEN
RETURN(TRUE);
ELSE
  IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CURVE_REPLICA'
      IN TYPEOF(crv) THEN
RETURN(valid_wireframe_edge_curve(crv\curve_replica.parent_curve));
  ELSE
    IF
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'OFFSET_CURVE_3D'
        IN TYPEOF(crv) THEN
RETURN(valid_wireframe_edge_curve(crv\offset_curve_3d.
basis_curve));
    END_IF;
  END_IF;
END_IF;
RETURN(FALSE);

END_FUNCTION; -- valid_wireframe_edge_curve

```

```

FUNCTION valid_wireframe_vertex_point(
  pnt: point
): BOOLEAN;
  IF
    'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
    'CARTESIAN_POINT'
      IN TYPEOF(pnt) THEN
RETURN(TRUE);
  ELSE
    IF
      'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
      'POINT_REPLICA'
        IN TYPEOF(pnt) THEN
RETURN(valid_wireframe_vertex_point(pnt\point_replica.parent_pt));
    END_IF;
  END_IF;
RETURN(FALSE);

```

```

END_FUNCTION; -- valid_wireframe_vertex_point

FUNCTION vector_difference(
    arg1, arg2: vector_or_direction
): vector;

LOCAL
    ndim    : INTEGER;
    mag2     : REAL;
    mag1     : REAL;
    mag      : REAL;
    res      : direction;
    vec1     : direction;
    vec2     : direction;
    result   : vector;
END_LOCAL;
IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) OR (arg1.dim <> arg2.dim)
    THEN
        RETURN(?);
    ELSE
        BEGIN
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'VECTOR'
                IN TYPEOF(arg1) THEN
                mag1 := arg1.magnitude;
                vec1 := arg1.orientation;
            ELSE
                mag1 := 1;
                vec1 := arg1;
            END_IF;
            IF
                'ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN.' +
                'VECTOR'
                IN TYPEOF(arg2) THEN
                mag2 := arg2.magnitude;
                vec2 := arg2.orientation;
            ELSE
                mag2 := 1;
                vec2 := arg2;
            END_IF;
            vec1 := normalise(vec1);
            vec2 := normalise(vec2);
            ndim := SIZEOF(vec1.direction_ratios);
            mag := 0;
            REPEAT i := 1 TO ndim BY 1;
                res.direction_ratios[i] := (mag1 * vec1.direction_ratios[i]) - (
                    mag2 * vec2.direction_ratios[i]);
                mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
            END_REPEAT;
            IF mag > 0 THEN

```

```
        result.magnitude := SQRT(mag);
        result.orientation := res;
    ELSE
        result.magnitude := 0;
        result.orientation := vec1;
    END_IF;
END;
END_IF;
result.name := '';
RETURN(result);

END_FUNCTION; -- vector_difference

END_SCHEMA; -- electronic_assembly_interconnect_and_packaging_design

(*
```

Annex B (normative) Short names of entities

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 – Short names of entities

Entity names	Short names
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_METHOD_RELATIONSHIP	ACMTRL
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
ADD_DESIGN_OBJECT_ASSIGNMENT	ADOA
ADD_DESIGN_OBJECT_REQUEST_ASSIGNMENT	ADORA
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ALTERNATE_PRODUCT_RELATIONSHIP	ALPRRL
AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT	AOSMWU
AMOUNT_OF_SUBSTANCE_UNIT	AOSU
ANALYTICAL_MODEL	ANLMDL
ANALYTICAL_MODEL_PORT	ANMDPR
ANALYTICAL_REPRESENTATION	ANLRPR
ANGULARITY_TOLERANCE	ANGTLR
ANGULAR_DIMENSION_WITH_ORIENTATION	ADWO
ANGULAR_LOCATION	ANGLCT
ANGULAR_SIZE	ANGSZ
ANNOTATION_CURVE_OCCURRENCE	ANCROC
ANNOTATION_FILL_AREA	ANFLAR

Table B.1 – Short names of entities (Continued)

Entity names	Short names
ANNOTATION_FILL_AREA_OCCURRENCE	AFAO
ANNOTATION_OCCURRENCE	ANNOCC
ANNOTATION_SYMBOL	ANNSYM
ANNOTATION_SYMBOL_OCCURRENCE	ANSYOC
ANNOTATION_TEXT	ANNTXT
ANNOTATION_TEXT_CHARACTER	ANTXCH
ANNOTATION_TEXT_OCCURRENCE	ANTXOC
APEX	APEX
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPLIED_ACTION_ASSIGNMENT	APACAS
APPLIED_APPROVAL_ASSIGNMENT	APAPAS
APPLIED_CERTIFICATION_ASSIGNMENT	APCRAS
APPLIED_CLASSIFICATION_ASSIGNMENT	APCLAS
APPLIED_CONTRACT_ASSIGNMENT	APCNAS
APPLIED_DATE_AND_TIME_ASSIGNMENT	ADATA
APPLIED_DATE_ASSIGNMENT	APDTAS
APPLIED_DOCUMENT_REFERENCE	APDCRF
APPLIED_GROUP_ASSIGNMENT	APGRAS
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA
APPLIED_PERSON_ASSIGNMENT	APPRAS
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ASCA
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
APPROVAL_DATE_TIME	APDTTM
APPROVAL_PERSON_ORGANIZATION	APPROR
APPROVAL_RELATIONSHIP	APPRLT
APPROVAL_ROLE	APPRL
APPROVAL_STATUS	APPSTT
AREA_MEASURE_WITH_UNIT	AMWU
AREA_UNIT	ARUNT
ARRAY_PLACEMENT_GROUP_COMPONENT_DEFINITION	APGCD
ARRAY_PLACEMENT_GROUP_COMPONENT_SHAPE_ASPECT	APGCSA
ASSEMBLY_BOND_DEFINITION	ASBNDF

Table B.1 – Short names of entities (Continued)

Entity names	Short names
ASSEMBLY_COMPONENT_USAGE	ASCMUS
ASSEMBLY_COMPONENT_USAGE_SUBSTITUTE	ACUS
ASSEMBLY_DEFINITION	ASSDFN
ASSEMBLY_GROUP_COMPONENT_DEFINITION	AGCD
ASSEMBLY_GROUP_COMPONENT_SHAPE_ASPECT	AGCSA
ASSEMBLY_JOINT	ASSJNT
ASSEMBLY_MODULE_TERMINAL	ASMDTR
AUXILIARY_CHARACTERISTIC_DIMENSION_REPRESENTATION	ACDR
AXIS1_PLACEMENT	AX1PLC
AXIS2_PLACEMENT_2D	A2PL2D
AXIS2_PLACEMENT_3D	A2PL3D
BARE_DIE	BRD
BARE_DIE_TERMINAL	BRDTR
BEZIER_CURVE	BZRCRV
BEZIER_SURFACE	BZRSRF
BLOCK	BLOCK
BOND_CATEGORY	BNDCTG
BOOLEAN_RESULT	BLNRSL
BOUNDARY_CURVE	BNDCR
BOUNDED_CURVE	BNDCRV
BOUNDED_PCURVE	BNDPCR
BOUNDED_SURFACE	BNDSRF
BOUNDED_SURFACE_CURVE	BNSRCR
BOXED_HALF_SPACE	BXHLSP
BOX_DOMAIN	BXDMN
BREP_WITH_VOIDS	BRWTVD
BUS_STRUCTURAL_DEFINITION	BSSTDF
B_SPLINE_CURVE	BSPCR
B_SPLINE_CURVE_WITH_KNOTS	BSCWK
B_SPLINE_SURFACE	BSPSR
B_SPLINE_SURFACE_WITH_KNOTS	BSSWK
CALENDAR_DATE	CLNDT
CARTESIAN_POINT	CRTPNT
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
CARTESIAN_TRANSFORMATION_OPERATOR_2D	CTO2
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3

Table B.1 – Short names of entities (Continued)

Entity names	Short names
CENTRE_OF_SYMMETRY	CNOFSY
CERTIFICATION	CRTFCT
CERTIFICATION_ASSIGNMENT	CRTASS
CERTIFICATION_TYPE	CRTTYP
CHANGE	CHANGE
CHANGE_FROM_DESIGN_OBJECT_ASSIGNMENT	CFDOA
CHANGE_FROM_DESIGN_OBJECT_REQUEST_ASSIGNMENT	CFDORA
CHANGE_REQUEST	CHNRQS
CHANGE_TO_DESIGN_OBJECT_ASSIGNMENT	CTDOA
CHANGE_TO_DESIGN_OBJECT_REQUEST_ASSIGNMENT	CTDORA
CHARACTERIZED_OBJECT	CHROBJ
CHARACTERIZED_PRODUCT_CATEGORY	CHPRCT
CHARACTER_GLYPH_FONT_USAGE	CGFU
CHARACTER_GLYPH_STYLE_OUTLINE	CGO
CHARACTER_GLYPH_STYLE_STROKE	CGS
CHARACTER_GLYPH_SYMBOL	CHGLSY
CHARACTER_GLYPH_SYMBOL_OUTLINE	CGSO
CIRCLE	CIRCLE
CIRCULAR_RUNOUT_TOLERANCE	CRRNTL
CLOSED_SHELL	CLSSHL
COLOUR	COLOUR
COLOUR_RGB	CLRRGB
COLOUR_SPECIFICATION	CLRSPC
COMPONENT_DEFINITION	CMPDFN
COMPONENT_FUNCTIONAL_TERMINAL	CMFNTR
COMPONENT_FUNCTIONAL_UNIT	CMFNUN
COMPONENT_INTERFACE_TERMINAL	CMINTR
COMPONENT_LOCATION	CMPLCT
COMPONENT_SHAPE_ASPECT	CMS1
COMPONENT_TERMINAL	CMPTRM
COMPOSITE_ARRAY_SHAPE_ASPECT	CASA
COMPOSITE_CURVE	CMPCRV
COMPOSITE_CURVE_ON_SURFACE	CCOS
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_GROUP_SHAPE_ASPECT	CGSA
COMPOSITE_SHAPE_ASPECT	CMSHAS

Table B.1 – Short names of entities (Continued)

Entity names	Short names
COMPOSITE_UNIT_SHAPE_ASPECT	CUSA
CONCENTRICITY_TOLERANCE	CNCTLR
CONFIGURATION_DESIGN	CNFDSG
CONFIGURATION_EFFECTIVITY	CNFEFF
CONFIGURATION_ITEM	CNFITM
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_EDGE_SET	CNEDST
CONNECTED_FACE_SET	CNFCST
CONNECTIVITY_SUB_STRUCTURE	CNSBST
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	CDSR
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONTRACT	CNTRCT
CONTRACT_ASSIGNMENT	CNTASS
CONTRACT_TYPE	CNTTYP
CONVERSION_BASED_UNIT	CNBSUN
COORDINATED_REPRESENTATION_ITEM	CRRPIT
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
CSG_SHAPE_REPRESENTATION	CSSHRP
CSG_SOLID	CSGSLD
CURVE	CURVE
CURVE_BOUNDED_SURFACE	CRBNSR
CURVE_DIMENSION	CRVDMN
CURVE_REPLICA	CRVRPL
CURVE_STYLE	CRVSTY
CURVE_STYLE_FONT	CRSTFN
CURVE_STYLE_FONT_AND_SCALING	CSFAS
CURVE_STYLE_FONT_PATTERN	CSFP
CURVE_STYLE_WITH_ENDS_AND_CORNERS	CSWEAC
CURVE_STYLE_WITH_EXTENSION	CSWE
CYLINDRICAL_SURFACE	CYLSRF
DATA_ENVIRONMENT	DTENV
DATE	DATE
DATED_EFFECTIVITY	DTDEFF
DATE_AND_TIME	DTANTM
DATE_AND_TIME_ASSIGNMENT	DATA

Table B.1 – Short names of entities (Continued)

Entity names	Short names
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATE_TIME_ROLE	DTMRL
DATUM_REFERENCE_FRAME	DTRFFR
DATUM_SYSTEM	DTMSYS
DATUM_SYSTEM_BASED_DIMENSIONAL_LOCATION	DSBDL
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_PCURVE	DGNPCR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR
DELETE_DESIGN_OBJECT_ASSIGNMENT	DDOA
DELETE_DESIGN_OBJECT_REQUEST_ASSIGNMENT	DDORA
DERIVED_SHAPE_ASPECT	DRSHAS
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
DESIGN_MAKE_FROM_RELATIONSHIP	DMFR
DESIGN_OBJECT	DSGOBJ
DEVICE_TERMINAL_MAP	DVTRMP
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	DMCHRP
DIMENSIONAL_EXPONENTS	DMNEXP
DIMENSIONAL_LOCATION	DMNLCT
DIMENSIONAL_LOCATION_WITH_DIRECTION	DLWD
DIMENSIONAL_LOCATION_WITH_PATH	DLWP
DIMENSIONAL_SIZE	DMNSZ
DIMENSIONAL_SIZE_PROPERTY	DMSZPR
DIRECTED_ACTION	DRCACT
DIRECTED_DIMENSIONAL_LOCATION	DRDMLC
DIRECTION	DRCTN
DISCRETE_SHIELD	DSCSHL
DOCUMENT	DCMNT
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DOCUMENT_WITH_CLASS	DCWTCL
EDGE	EDGE

Table B.1 – Short names of entities (Continued)

Entity names	Short names
EDGE_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION	EB2WSR
EDGE_BASED_WIREFRAME_MODEL	EBWM
EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION	EBWSR
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EE_SPECIFICATION	ESPC
EFFECTIVITY	EFFCTV
ELECTRICAL_NETWORK	ELCNTW
ELECTRIC_CURRENT_MEASURE_WITH_UNIT	ECMWU
ELECTRIC_CURRENT_UNIT	ELCRUN
ELEMENTARY_SURFACE	ELMSRF
ELLIPSE	ELLPS
EVALUATED_DEGENERATE_PCURVE	EVDGPC
EXECUTED_ACTION	EXCACT
EXPANDED_UNCERTAINTY	EXPUNC
EXTENSION	EXTNSN
EXTERNALLY_DEFINED_ASSEMBLY_DEFINITION	EDAD
EXTERNALLY_DEFINED_BARE_DIE	EDBD
EXTERNALLY_DEFINED_CURVE_FONT	EDCF
EXTERNALLY_DEFINED_FUNCTIONAL_UNIT	EDFU
EXTERNALLY_DEFINED_HATCH_STYLE	EDHS
EXTERNALLY_DEFINED_INTERCONNECT_DEFINITION	EDID
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_PACKAGE	EXDFPC
EXTERNALLY_DEFINED_PACKAGED_CONNECTOR	EDP0
EXTERNALLY_DEFINED_PACKAGED_PART	EDPP
EXTERNALLY_DEFINED_PHYSICAL_UNIT	EDPU
EXTERNALLY_DEFINED_REPRESENTATION_ITEM	EDRI
EXTERNALLY_DEFINED_SYMBOL	EXDFSY
EXTERNALLY_DEFINED_TEXT_FONT	EDTF
EXTERNALLY_DEFINED_TILE	EXDFTL
EXTERNALLY_DEFINED_TILE_STYLE	EDTS
EXTERNAL_DEFINITION	EXTDFN
EXTERNAL_SOURCE	EXTSRC
EXTERNAL_SOURCE_RELATIONSHIP	EXSRRL
EXTRUDED_AREA_SOLID	EXARSL

Table B.1 – Short names of entities (Continued)

Entity names	Short names
FABRICATION_JOINT	FBRJNT
FACE	FACE
FACETED_BREP	FCTBR
FACETED_BREP_SHAPE_REPRESENTATION	FBSR
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SURFACE	FCSRF
FIDUCIAL	FDCL
FIDUCIAL_PART_FEATURE	FDPRFT
FIDUCIAL_STRATUM_FEATURE	FDSTFT
FILL_AREA_STYLE	FLARST
FILL_AREA_STYLE_COLOUR	FASC
FILL_AREA_STYLE_HATCHING	FASH
FILL_AREA_STYLE_TILES	FAST
FILL_AREA_STYLE_TILE_COLOURED_REGION	FASTCR
FILL_AREA_STYLE_TILE_CURVE_WITH_STYLE	FASTCW
FILL_AREA_STYLE_TILE_SYMBOL_WITH_STYLE	FASTSW
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
FUNCTIONAL_TERMINAL_GROUP	FNTRGR
FUNCTIONAL_UNIT	FNCUNT
FUNCTIONAL_UNIT_TERMINAL_DEFINITION	FUTD
GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION	GB2WR
GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION	GBSSR
GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION	GBWSR
GEOMETRIC_ALIGNMENT	GMTALG
GEOMETRIC_CURVE_SET	GMCRST
GEOMETRIC_INTERSECTION	GMTINT
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_SET	GMTST
GEOMETRIC_TOLERANCE	GMTTLR
GEOMETRIC_TOLERANCE_GROUP	GMTLGR
GEOMETRIC_TOLERANCE_RELATIONSHIP	GMTLRL

Table B.1 – Short names of entities (Continued)

Entity names	Short names
GEOMETRIC_TOLERANCE_WITH_SPECIFIED_DATUM_SYSTEM	GTWSDS
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUPED_REQUIREMENTS_PROPERTY	GRRQPR
GROUP_ASSIGNMENT	GRPASS
GROUP_PRODUCT_DEFINITION	GRPRDF
GROUP_RELATIONSHIP	GRPRLT
GROUP_SHAPE_ASPECT	GRSHAS
HALF_SPACE_SOLID	HLSPSL
HYPERBOLA	HYPRBL
INTEGRAL_SHIELD	INTSHL
INTERCONNECT_DEFINITION	INTDFN
INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY	IMDOC
INTERCONNECT_MODULE_STRATUM_BASED_TERMINAL	IMSBT
INTERCONNECT_MODULE_TERMINAL	INMDTR
INTERFACED_GROUP_COMPONENT_DEFINITION	IGCD
INTERFACE_COMPONENT_DEFINITION	INCMDF
INTERFACE_MOUNTED_JOIN	INMNJN
INTERSECTION_CURVE	INTCRV
INTER_STRATUM_FEATURE	INSTFT
ITEM_DEFINED_TRANSFORMATION	ITDFTR
JOIN_SHAPE_ASPECT	JNSHAS
LAND	LAND
LAND_PHYSICAL_TEMPLATE	LNPHTM
LAND_TEMPLATE_TERMINAL	LNTMTR
LAYER	LAYER
LAYER_CONNECTION_POINT	LYCNPN
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIBRARY_DEFINED_ASSEMBLY_DEFINITION	LDAD
LIBRARY_DEFINED_BARE_DIE	LDBD
LIBRARY_DEFINED_FUNCTIONAL_UNIT	LDFU
LIBRARY_DEFINED_INTERCONNECT_DEFINITION	LDID
LIBRARY_DEFINED_MODEL	LBDFMD
LIBRARY_DEFINED_PACKAGE	LBDFPC

Table B.1 – Short names of entities (Continued)

Entity names	Short names
LIBRARY_DEFINED_PACKAGED_CONNECTOR	LDPC
LIBRARY_DEFINED_PACKAGED_PART	LDPP
LIBRARY_DEFINED_PHYSICAL_UNIT	LDPU
LINE	LINE
LINEAR_PROFILE_TOLERANCE	LNPRTL
LOCAL_TIME	LCLTM
LOOP	LOOP
LOT_EFFECTIVITY	LTEFF
LUMINOUS_INTENSITY_MEASURE_WITH_UNIT	LIMWU
LUMINOUS_INTENSITY_UNIT	LMINUN
MAKE_FROM_CONNECTIVITY_RELATIONSHIP	MFCR
MAKE_FROM_FEATURE_RELATIONSHIP	MFFR
MAKE_FROM_USAGE_OPTION	MFUO
MANIFOLD_SOLID_BREP	MNSLBR
MANIFOLD_SURFACE_SHAPE_REPRESENTATION	MSSR
MAPPED_ITEM	MPPITM
MASS_MEASURE_WITH_UNIT	MMWU
MASS_UNIT	MSSUNT
MATERIAL_DESIGNATION	MTRDSG
MATERIAL_DESIGNATION_CHARACTERIZATION	MTDSCH
MATERIAL_ELECTRICAL_CONDUCTIVITY_CATEGORY	MECC
MATERIAL_PROPERTY	MTRPRP
MATERIAL_PROPERTY_REPRESENTATION	MTPRRP
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MINIMALLY_DEFINED_BARE_DIE_TERMINAL	MDBDT
MODEL_PARAMETER	MDLPRM
MODIFIED_GEOMETRIC_TOLERANCE	MDGMTL
MOUNTING_RESTRICTION_AREA	MNRSAR
NAMED_UNIT	NMDUNT
NETWORK_NODE_DEFINITION	NTNDDF
NEXT_ASSEMBLY_USAGE_OCCURRENCE	NAUO
OFFSET_CURVE_2D	OFCR2D
OFFSET_CURVE_3D	OFCR3D
OFFSET_SURFACE	OFFSRF

Table B.1 – Short names of entities (Continued)

Entity names	Short names
ONE_DIRECTION_REPEAT_FACTOR	ODRF
OPEN_SHELL	OPNSHL
OPPOSING_BOUNDARY_DIMENSIONAL_SIZE	OBDS
ORGANIZATION	ORGNZT
ORGANIZATIONAL_ADDRESS	ORGADD
ORGANIZATIONAL_PROJECT	ORGPRJ
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_RELATIONSHIP	ORGRLT
ORGANIZATION_ROLE	ORGRL
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
OUTER_BOUNDARY_CURVE	OTBNCR
PACKAGE	PCKG
PACKAGED_COMPONENT	PCKCMP
PACKAGED_CONNECTOR	PCKCNN
PACKAGED_CONNECTOR_TERMINAL_RELATIONSHIP	PCTR
PACKAGED_PART	PCKPRT
PACKAGED_PART_TERMINAL	PCPRTR
PACKAGE_BODY	PCKBDY
PACKAGE_TERMINAL	PCKTRM
PARABOLA	PRBL
PARALLELISM_TOLERANCE	PRLTLR
PARAMETER_ASSIGNMENT_REPRESENTATION	PRASRP
PARAMETER_TYPE	PRMTYP
PARAMETRIC_REPRESENTATION_CONTEXT	PRRPCN
PART_CONNECTED_TERMINALS_DEFINITION	PCTD
PART_MOUNTING_FEATURE	PRMNFT
PART_TEMPLATE_DEFINITION	PRTMDF
PART_TEXT_TEMPLATE_DEFINITION	PTTD
PART_TOOLING_FEATURE	PRTLFT
PASSAGE_TECHNOLOGY	PSSTCH
PATH	PATH
PCURVE	PCURVE

Table B.1 – Short names of entities (Continued)

Entity names	Short names
PERPENDICULARITY_TOLERANCE	PRPTLR
PERSON	PERSON
PERSONAL_ADDRESS	PRSADD
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSON_ASSIGNMENT	PRSASS
PERSON_ROLE	PRSRL
PHYSICAL_CONNECTIVITY_DEFINITION	PHCNDF
PHYSICAL_CONNECTIVITY_ELEMENT	PHCNEL
PHYSICAL_NETWORK	PHYNTW
PHYSICAL_UNIT	PHYUNT
PHYSICAL_UNIT_DATUM	PHUNDT
PHYSICAL_UNIT_DATUM_FEATURE	PUDF
PHYSICAL_UNIT_DATUM_TARGET	PUDT
PHYSICAL_UNIT_DATUM_TARGET_SET	PUDTS
PHYSICAL_UNIT_GEOMETRIC_TOLERANCE	PUGT
PLACEMENT	PLCMNT
PLANAR_EXTENT	PLNEXT
PLANE	PLANE
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN
PLATED_INTER_STRATUM_FEATURE	PISF
PLATED_PASSAGE	PLTPSS
PLUS_MINUS_TOLERANCE	PLMNTL
POINT	POINT
POINT_ON_CURVE	PNONCR
POINT_ON_SURFACE	PNONSR
POINT_REPLICA	PNTRPL
POLYLINE	PLYLN
POLY_LOOP	PLYLP
POSITIONAL_BOUNDARY	PSTBND
POSITIONAL_BOUNDARY_MEMBER	PSBNMM
POSITION_TOLERANCE	PSTTLR
PRECISION_QUALIFIER	PRCQLF
PRESENTATION_REPRESENTATION	PRSRPR

Table B.1 – Short names of entities (Continued)

Entity names	Short names
PRESENTATION_STYLE_ASSIGNMENT	PRSTAS
PRESENTATION_STYLE_BY_CONTEXT	PSBC
PRE_DEFINED_CURVE_FONT	PDCF
PRE_DEFINED_ITEM	PRDFIT
PRIMARY_ORIENTATION_FEATURE	PRORFT
PRIMARY_REFERENCE_TERMINAL	PRRFTR
PRINTED_COMPONENT	PRNCMP
PRINTED_PART_CROSS_SECTION_TEMPLATE_TERMINAL	PPCSTT
PRINTED_PART_TEMPLATE_CONNECTED_TERMINALS_DEFINITION	PPTCTD
PRINTED_PART_TEMPLATE_TERMINAL	PPTT
PRINTED_PART_TEMPLATE_TERMINAL_CONNECTION_ZONE_CATEGORY	PPTTCZ
PROBE_ACCESS_AREA	PRACAR
PRODUCT	PRDCT
PRODUCT_CATEGORY	PRDCTG
PRODUCT_CATEGORY_RELATIONSHIP	PRCTRL
PRODUCT_CONCEPT	PRDCNC
PRODUCT_CONCEPT_CONTEXT	PRCNCN
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_EFFECTIVITY	PRDFEF
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE	PDFWSS
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_USAGE	PRDFUS
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP	PMCR
PRODUCT_RELATED_CHARACTERIZED_PRODUCT_CATEGORY	PRCPC
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PROJECTED_ZONE_DEFINITION	PRZNDF
PROMISSORY_USAGE_OCCURRENCE	PRUSOC

Table B.1 – Short names of entities (Continued)

Entity names	Short names
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
QUALITATIVE_UNCERTAINTY	QLTUNC
QUANTIFIED_ASSEMBLY_COMPONENT_USAGE	QACU
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RECTANGULAR_COMPOSITE_SURFACE	RCCMSR
RECTANGULAR_TRIMMED_SURFACE	RCTRSR
REPARAMETRISED_COMPOSITE_CURVE_SEGMENT	RCCS
REPRESENTATION	RPRSNT
REPRESENTATION_CONTEXT	RPRCNT
REPRESENTATION_ITEM	RPRITM
REPRESENTATION_MAP	RPRMP
REPRESENTATION_RELATIONSHIP	RPRRLT
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION	RRWT
REQUIREMENTS_PROPERTY	RQRPRP
REQUIREMENTS_PROPERTY_GROUP	RQPRGR
REQUIREMENT_ALLOCATION_GROUP	RQALGR
REVOLVED_AREA_SOLID	RVARSL
RIGHT_ANGULAR_WEDGE	RGANWD
RIGHT_CIRCULAR_CONE	RGCRCN
RIGHT_CIRCULAR_CYLINDER	RGCRCY
ROUTED_PRINTED_COMPONENT	RTPRCM
RULE_ACTION	RLACT
RULE_DEFINITION	RLDFN
RULE_FUNCTION_DEFINITION	RLFNDF
RULE_REPLACEMENT_FROM_ASSIGNMENT	RRFA
RULE_REPLACEMENT_FROM_REQUEST_ASSIGNMENT	RRFRA
RULE_REPLACEMENT_TO_ASSIGNMENT	RRTA
RULE_REPLACEMENT_TO_REQUEST_ASSIGNMENT	RRTRA
RULE_SET	RLST

Table B.1 – Short names of entities (Continued)

Entity names	Short names
RUNOUT_ZONE_DEFINITION	RNZNDF
RUNOUT_ZONE_ORIENTATION	RNZNOR
RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION	RZORD
SEAM_CURVE	SMCRV
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SEQUENTIAL_LAMINATE_PASSAGE_BASED_FABRICATION_- JOINT	SLPBFJ
SERIAL_NUMBERED_EFFECTIVITY	SRNMEF
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_DERIVING_RELATIONSHIP	SADR
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_DIMENSION_REPRESENTATION	SHDMRP
SHAPE_MODIFICATION	SHPMDF
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_RELATIONSHIP	SHRPRL
SHELL_BASED_2D_WIREFRAME_SHAPE_REPRESENTATION	SB2WSR
SHELL_BASED_SURFACE_MODEL	SBSM
SHELL_BASED_WIREFRAME_MODEL	SBWM
SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION	SBWSR
SIGNAL	SIGNAL
SIGNAL_CATEGORY	SGNCTG
SI_UNIT	SUNT
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
SOLID_ANGLE_UNIT	SLANUN
SOLID_CURVE_FONT	SLCRFN
SOLID_MODEL	SLDMDL
SOLID_REPLICA	SLDRPL
SPECIFIED_HIGHER_USAGE_OCCURRENCE	SHUO
SPHERE	SPHERE
SPHERICAL_SURFACE	SPHSRF
STANDARD_UNCERTAINTY	STNUNC
START_REQUEST	STRRQS
START_WORK	STRWRK

Table B.1 – Short names of entities (Continued)

Entity names	Short names
STATISTICAL_DIMENSIONAL_LOCATION	STDMLC
STATISTICAL_DIMENSIONAL_SIZE	STDMSZ
STATISTICAL_GEOMETRIC_TOLERANCE	STGMTL
STRATUM	STRTM
STRATUM_CONCEPT_RELATIONSHIP	STCNRL
STRATUM_FEATURE	STRFTR
STRATUM_SURFACE	STRSRF
STRATUM_TECHNOLOGY	STRTCH
STYLED_ITEM	STYITM
SUPPLIED_PART_RELATIONSHIP	SPPRRL
SURFACE	SRFC
SURFACE_CURVE	SRFCRV
SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PATCH	SRFPTC
SURFACE_PROFILE_TOLERANCE	SRPRTL
SURFACE_REPLICA	SRFRPL
SWEPT_AREA_SOLID	SWARSL
SWEPT_SURFACE	SWPSRF
SYMBOL_REPRESENTATION	SYMRPR
SYMBOL_REPRESENTATION_MAP	SYRPMP
SYMBOL_TARGET	SYMTRG
SYMMETRIC_SHAPE_ASPECT	SYSHAS
SYMMETRY_TOLERANCE	SYMTLR
TANGENT	TNGNT
TEXT_FONT	TXTFNT
TEXT_FONT_FAMILY	TXFNFM
TEXT_FONT_IN_FAMILY	TFIF
TEXT_LITERAL	TXTLTR
TEXT_LITERAL_WITH_ASSOCIATED_CURVES	TLWAC
TEXT_LITERAL_WITH_EXTENT	TLWE
TEXT_STRING_REPRESENTATION	TXSTRP
TEXT_STYLE	TXTSTY
TEXT_STYLE_FOR_DEFINED_FONT	TSFDF
THERMAL_COMPONENT	THRCMP
THERMAL_COMPONENT_SHAPE_ASPECT	TCSA

Table B.1 – Short names of entities (Concluded)

Entity names	Short names
THERMAL_FEATURE	THRFTTR
THERMAL_NETWORK	THRNTW
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	TTMWU
THERMODYNAMIC_TEMPERATURE_UNIT	THTMUN
TIME_MEASURE_WITH_UNIT	TMWU
TIME_UNIT	TMUNT
TOLERANCE_VALUE	TLRVL
TOLERANCE_ZONE	TLRZN
TOLERANCE_ZONE_BOUNDARY	TLZNBN
TOLERANCE_ZONE_DEFINITION	TLZNDF
TOLERANCE_ZONE_EXPLICIT_OPPOSING_BOUNDARY_SET	TZEOBS
TOLERANCE_ZONE_FORM	TLZNFR
TOLERANCE_ZONE_IMPLICIT_OPPOSING_BOUNDARY_SET	TZIOBS
TOPOLOGICAL_REPRESENTATION_ITEM	TPRPIT
TOROIDAL_SURFACE	TRDSRF
TORUS	TORUS
TOTAL_RUNOUT_TOLERANCE	TTRNTL
TRIMMED_CURVE	TRMCRV
TWO_DIRECTION_REPEAT_FACTOR	TDRF
TYPE_QUALIFIER	TYPQLF
UNCERTAINTY_MEASURE_WITH_UNIT	UMWU
UNCERTAINTY_QUALIFIER	UNCQLF
UNIFORM_CURVE	UNFCRV
UNIFORM_SURFACE	UNFSRF
VECTOR	VECTOR
VERSIONED_ACTION_REQUEST	VRACRQ
VERTEX	VERTEX
VERTEX_LOOP	VRTLP
VERTEX_POINT	VRTPNT
VERTEX_SHELL	VRTSHL
VIEWING_PLANE	VWNPLN
VOLUME_MEASURE_WITH_UNIT	VMWU
VOLUME_UNIT	VLMUNT
WIRE_SHELL	WRSHL

Annex C
(normative)
Implementation method specific requirements

The implementation method defines what types of exchange behaviour are required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and in the AIM defined in annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'electronic_assembly_interconnect_and_packaging_design'.

Exchange between implementations shall be limited to implementations which have implemented the same (or none) 2D shape representation conformance classes unless 2D to 3D conversion is employed. Exchange between implementations shall be limited to implementations which have implemented the same (or none) 3D shape representation conformance classes unless 2D to 3D conversion is employed.

Annex D (normative)

Protocol Implementation Conformance Statement (PICS) proforma

The Protocol Implementation Conformance Statement (PICS) proforma is supplied for completion by the person or organization (the client) requesting conformance testing. Its purpose is to ascertain the scope of claimed conformance to a particular application protocol by an implementation under test (IUT) using a defined implementation method. Through the completion of this form, the PICS Proforma becomes a PICS.

The information contained in the PICS is used to configure an appropriate executable test suite for use by the client.

Conformance classes and the appropriate relationships between conformance classes for this part of ISO 10303 are detailed in clause 6.

Questions:

1. Please provide an identifier for the product or system for which conformance is claimed:

Product name and current version number: _____

2. Please indicate the implementation method chosen:

☐ ISO 10303-21 Exchange Structure – preprocessor

Preprocessor name and current version number: _____

☐ ISO 10303-21 Exchange Structure – postprocessor

Postprocessor name and current version number: _____

☐ ISO 10303-22 Exchange Interface – processor

Interface Processor name and current version number: _____

3. Please indicate the classes for which conformance is claimed:

☐ Class 1 - Device Functional and Physical Characteristics

☐ Class 2 - Interconnect Technology Constraints

☐ Class 3 - Assembly Technology Constraints

☐ Class 4 - Assembly Functional Requirements

- _ Class 5 - Assembly Physical Requirements
- _ Class 6 - Interconnect Functional Requirements
- _ Class 7 - Interconnect Physical Requirements
- _ Class 8 - Assembly Physical Design
- _ Class 9 - Interconnect Design
- _ Class 10 - Interconnect Design (Microwave)
- _ Class 11 - Geometric Dimensioning and Tolerancing
- _ Class 12 - Product Rule
- _ Class 13 - Functional Decomposition
- _ Class 14 - Package Functional and Physical Characterization
- _ Class 15 - Geometrically Bounded Surface Model
- _ Class 16 - Wireframe Model with Topology
- _ Class 17 - Advanced Boundary Representation
- _ Class 18 - Constructive Solid Geometry
- _ Class 19 - Extruded solid
- _ Class 20 - Geometrically Bounded 2d Wireframe Model
- _ Class 21 - Wireframe 2d Model with Topology
- _ Class 22 - Curve 2d
- _ Class 23 - Basic Curve 2d
- _ Class 24 - Laminate Assembly Design
- _ Class 25 - Connection Zone Based Model Extraction
- _ Class 26 - Functional Specification
- _ Class 27 - Physical Unit Physical Characterization

- _ Class 28 - Packaged Part White Box Model
- _ Class 29 - Printed Part Functional and Physical Characterization
- _ Class 30 - Open Shell Model
- _ Class 31 - Assembly Functional Interface Requirements
- _ Class 32 - Assembly Physical Interface Requirements
- _ Class 33 - Assembly Requirements Allocation
- _ Class 34 - Interconnect Requirement Allocation
- _ Class 35 - Functional Requirement Allocation
- _ Class 36 - Product Requirement Allocation
- _ Class 37 - Datum Difference Based Model Definition
- _ Class 38 - Datum Difference Based Characteristic
- _ Class 39 - Design Management

4. Please indicate whether Date only or Date and Time to the second is supported:

Date only _____

Date and Time to the Second _____

User selects between the above two choices _____

Annex E (normative) Information object registration

E.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(210) version(1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

E.2 Schema identification

To provide for unambiguous identification of the schema specifications given in this application protocol `electronic_assembly_interconnect_and_packaging_design` in an open information system, object identifiers are assigned as follows:

{ iso standard 10303 part(210) version(1) object(1)
`electronic-assembly-interconnect-and-packaging-design(1)` }

is assigned to the `electronic_assembly_interconnect_and_packaging_design` expanded schema (see annex A).

{ iso standard 10303 part(210) version(1) object(1)
`electronic-assembly-interconnect-and-packaging-design(2)` }

is assigned to the `electronic_assembly_interconnect_and_packaging_design` short form schema (see 5.2).

The meaning of these values is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F (normative) Application interpreted constructs

An application interpreted construct (AIC) provides a logical grouping of interpreted constructs that supports a specific functionality for the usage of product data across multiple application contexts. An interpreted construct is a common interpretation of the integrated resources that supports shared information requirements among application protocols.

NOTE SC4 resolution 286 (Kobe June 1996) states that

- an AIC shall be published as a separate part (500-series document)
- APs shall reference AICs only as a normative reference
- AICs shall be at the same or advanced voting status as an AP in which it is being used at the the time of the AP's DIS or FDIS ballot.

As a result of this resolution, ISO 10303-210's annex sequence shall differ from the annex sequence defined by the ISO TC184/SC4/N537 *Supplementary directives for the drafting and presentation of ISO 10303*. Annex F of this part of ISO 10303 contains text copied verbatim from the AIC ISO 10303-507 (CD) and the AIC ISO 10303-509 (CD).

F.1 Geometrically bounded surface

F.1.1 Scope

This application interpreted construct specifies the interpretation of the integrated resources to satisfy requirements for the description of geometric shapes by means of geometrically bounded surface models.

The following are within the scope of this AIC:

- 3D points;
- points defined in the parameter space of curves or surfaces;
- 3D curves;
- curves defined in the parameter space of surfaces;

NOTE Such curves are also known as pcurves or cons, which are acronyms for parametrised curve and curve on surface.

- the elementary curves, i.e., line, circle, ellipse, parabola, and hyperbola;
- intersection curves;
- polylines that consist of at least three points;

- surfaces;
- the elementary surfaces, i.e., plane, cylinder, cone, torus, and sphere;
- swept surfaces created by rotation or linear extrusion of a curve;
- sculptured curves and surfaces;
- trimming of curves and surfaces;
- composition of curves and surfaces;
- replication of curves, surfaces, and surface models;
- 3D offsets of curves and surfaces.

The following are outside the scope of this AIC:

- unbounded geometry;
- self-intersecting geometry;
- geometry in a 2D cartesian coordinate space;
- replication of points;
- topological entities.

F.1.2 EXPRESS short listing

This application interpreted construct provides a consistent set of geometric entities for the definition of surface models that consist of points, elementary or sculptured curves, and elementary or sculptured surfaces. Geometry shall be bounded; no topological entities are used for bounding.

EXPRESS specification:

```
*)  
SCHEMA aic_geometrically_bounded_surface;  
  
    USE FROM geometric_model_schema (  
        geometric_set);  
  
    USE FROM geometry_schema (  
        axis1_placement,  
        axis2_placement_2d,  
        axis2_placement_3d,
```

b_spline_curve,
b_spline_curve_with_knots,
b_spline_surface,
b_spline_surface_with_knots,
bezier_curve,
bezier_surface,
boundary_curve,
bounded_pcurve,
bounded_surface_curve,
cartesian_point,
cartesian_transformation_operator_3d,
circle,
composite_curve,
composite_curve_on_surface,
composite_curve_segment,
conical_surface,
curve,
curve_bounded_surface,
curve_replica,
cylindrical_surface,
degenerate_pcurve,
degenerate_toroidal_surface,
direction,
ellipse,
evaluated_degenerate_pcurve,
geometric_representation_context,
hyperbola,
intersection_curve,
line,
offset_curve_3d,
offset_surface,
outer_boundary_curve,
parabola,
pcurve,
plane,
point,
point_on_curve,
point_on_surface,
polyline,
quasi_uniform_curve,
quasi_uniform_surface,
rational_b_spline_curve,
rational_b_spline_surface,
rectangular_composite_surface,
rectangular_trimmed_surface,
reparametrised_composite_curve_segment,
seam_curve,
spherical_surface,
surface,
surface_curve,
surface_of_linear_extrusion,
surface_of_revolution,

```

surface_patch,
surface_replica,
swept_surface,
toroidal_surface,
trimmed_curve,
uniform_curve,
uniform_surface,
vector);

```

```

USE FROM product_property_representation_schema (
  shape_representation);

```

```

USE FROM representation_schema (
  definitional_representation,
  mapped_item,
  parametric_representation_context,
  representation,
  representation_map);

```

```
(*
```

NOTE 1 The schemas referenced above can be found in the following parts of ISO 10303:

geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
product_property_representation_schema	ISO 10303-41
representation_schema	ISO 10303-43

F.1.2.1 Fundamental concepts and assumptions

The following entities are intended to be independently instantiated in the application protocol schemas that use this AIC:

— `geometrically_bounded_surface_shape_representation`.

F.1.2.2 `aic_geometrically_bounded_surface` schema entity definitions

F.1.2.2.1 `geometrically_bounded_surface_shape_representation`

A **`geometrically_bounded_surface_shape_representation`** describes the shape or portions of the shape of a **product** using a surface model without topology.

NOTE 1 Entity **product** is not part of this AIC.

The highest level entity of this AIC is **`geometrically_bounded_surface_shape_representation`**. It is a **shape_representation** (see ISO 10303-41) consisting of **geometric_sets**. **Points**, **curves**, and **surfaces** may be contained in a **geometric_set** provided they are of the same dimensionality (see ISO 10303-42, rule **compatible_dimension**). A **`geometrically_bounded_surface_shape_representation`** shall include at least one **surface**. The geometric entities that are exclusively used to support the

definition of other geometric entities shall not themselves exist in the sets of **elements** of a **geometric_set**. All geometric entities shall be of dimensionality three except for two-dimensional geometry that is used for the purpose of defining **pcurves**. The use of one-dimensional **cartesian_points** is excluded.

All unbounded curves and surfaces shall be explicitly trimmed. The boundaries of curves shall be defined either by points on curves and explicit associations between these points and curves, or by parameter values. The boundaries of surfaces shall be defined either by curves on surfaces and explicit associations between these curves and surfaces, or by parameter values.

The **items** of a **geometrically_bounded_surface_shape_representation** may also be of type **mapped_item** (see ISO 10303-43) or **axis2_placement_3d**. These are used to assemble one or several **geometrically_bounded_surface_shape_representations** into one other **geometrically_bounded_surface_shape_representation**.

The WHERE-rules of this entity restrict the use of the entity data types that are imported from ISO 10303-42 and ISO 10303-43 according to the statements above. Most of these type checks and validations of constraints are specified in the following three functions:

- **gbsf_check_point;**
- **gbsf_check_curve;**
- **gbsf_check_surface.**

The three functions shall be applied to all **elements** of all **geometric_sets** in a **geometrically_bounded_surface_shape_representation**. The functions automatically assess all **points**, **curves**, and **surfaces** that are referenced by these **elements**. The functions, therefore, apply recursion.

EXAMPLE 1 A **pcurve** references both a **curve** and a **surface**. Function **gbsf_check_curve** validates not only the **pcurve**, but also its underlying geometry. It will, therefore, not only call itself, but also **gbsf_check_surface**.

NOTE 2 An application protocol that uses this part of ISO 10303 should explicitly require that the **shape_representation** entity may be instantiated as a **geometrically_bounded_surface_shape_representation**.

EXPRESS specification:

*)

```
ENTITY geometrically_bounded_surface_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1 : SIZEOF (QUERY (it <* SELF.items |
    NOT (SIZEOF ([ 'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET',
      'AIC_GEOMETRICALLY_BOUNDED_SURFACE.MAPPED_ITEM',
      'AIC_GEOMETRICALLY_BOUNDED_SURFACE.AXIS2_PLACEMENT_3D'] * TYPEOF
      (it)) = 1))) = 0;
```

```

WR2 : SIZEOF (QUERY (it <* SELF.items |
    SIZEOF ([ 'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET',
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.MAPPED_ITEM'] * TYPEOF
    (it)) = 1)) > 0;
WR3 : SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.MAPPED_ITEM' IN TYPEOF (it)) |
    NOT ('AIC_GEOMETRICALLY_BOUNDED_SURFACE.' +
    'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION'
    IN TYPEOF (mi\mapped_item.mapping_source.mapped_representation))))
    = 0;
WR4 : SIZEOF (QUERY (gs <* QUERY (it <* SELF.items |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET' IN TYPEOF (it)) |
    NOT (SIZEOF (QUERY (pnt <* QUERY (gsel <*
    gs\geometric_set.elements |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.POINT' IN TYPEOF (gsel)) |
    NOT (gbsf_check_point(pnt)))) = 0))) = 0;
WR5 : SIZEOF (QUERY (gs <* QUERY (it <* SELF.items |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET' IN TYPEOF (it)) |
    NOT (SIZEOF (QUERY (cv <* QUERY (gsel <*
    gs\geometric_set.elements |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.CURVE' IN TYPEOF (gsel)) |
    NOT (gbsf_check_curve(cv)))) = 0))) = 0;
WR6 : SIZEOF (QUERY (gs <* QUERY (it <* SELF.items |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET' IN TYPEOF (it)) |
    NOT (SIZEOF (QUERY (sf <* QUERY (gsel <*
    gs\geometric_set.elements |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.SURFACE' IN TYPEOF (gsel)) |
    NOT (gbsf_check_surface(sf)))) = 0))) = 0;
WR7 : SIZEOF (QUERY (gs <* QUERY (it <* SELF.items |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.GEOMETRIC_SET' IN TYPEOF (it)) |
    SIZEOF (QUERY (gsel <* gs\geometric_set.elements |
    'AIC_GEOMETRICALLY_BOUNDED_SURFACE.SURFACE'
    IN TYPEOF (gsel))) > 0)) > 0;

END_ENTITY;
(*

```

Formal propositions:

WR1: The **items** in a **geometrically_bounded_surface_shape_representation** shall be **geometric_sets**, **mapped_items**, or **axis2_placement_3ds**.

NOTE 3 **Axis2_placement_3d** is a valid **mapped_item.mapping_target**. To include another **representation** into the list of **items** of a **geometrically_bounded_surface_shape_representation** (see WR3 for valid **mapped_items**), the **mapped_item.mapping_source.mapping_origin** may be any entity that is geometrically founded in the **geometric_representation_context** of the **mapped_representation**. If this entity is an **axis2_placement_3d**, the operator that maps the **mapped_representation** into the **geometrically_bounded_surface_shape_representation** corresponds to a transformation matrix with only translation and rotation enabled. If a **cartesian_transformation_operator_3d** is used as **mapping_origin**, scaling and mirroring are possible.

WR2: At least one of the **items** in a **geometrically_bounded_surface_shape_representation** shall be either a **geometric_set** or a **mapped_item**.

WR3: If there is a **mapped_item** in a **geometrically_bounded_surface_shape_representation**, the **mapped_representation** of its **mapping_source** shall be a **geometrically_bounded_surface_shape_representation**.

WR4: Each **point** that is among the **elements** of a **geometric_set** that is one of the **items** of a **geometrically_bounded_surface_shape_representation** shall be a valid **point**.

WR5: Each **curve** that is among the **elements** of a **geometric_set** that is one of the **items** of a **geometrically_bounded_surface_shape_representation** shall be a valid **curve**.

WR6: Each **surface** that is among the **elements** of a **geometric_set** that is one of the **items** of a **geometrically_bounded_surface_shape_representation** shall be a valid **surface**.

WR7: At least one of the **elements** of at least one **geometric_set** that is among the **items** of a **geometrically_bounded_surface_shape_representation** shall be a **surface**.

Informal propositions:

IP1: A **b_spline_curve** shall not self-intersect.

IP2: A **composite_curve** shall not self-intersect.

IP3: An **offset_curve_3d** shall not self-intersect.

IP4: A **b_spline_surface** shall not self-intersect.

IP5: An **offset_surface** shall not self-intersect.

IP6: The geometric entities that are exclusively used to support the definition of other geometric entities shall not themselves exist in the sets of **elements** of a **geometric_set**.

F.1.2.3 aic_geometrically_bounded_surface function definitions

This section describes functions required to formulate constraints for the **aic_geometrically_bounded_surface** schema. These functions are used in the specification of the entity **geometrically_bounded_surface_shape_representation**.

F.1.2.3.1 gbsf_check_point

The **gbsf_check_point** function performs type checking for **points**. An instance is checked against all types of **points** that are valid in the context of a **geometrically_bounded_surface_shape_representation**. All related geometry, that is **curves** and **surfaces**, are also checked in the same way. When **curves** or **surfaces** are referenced, the functions **gbsf_check_curve** respectively **gbsf_check_surface** are called. The recursion within these functions terminates at entity types that do not reference any **points**, **curves**, or **surfaces**.

The following point types and their subtypes are within the scope of the **geometrically_bounded_surface_shape_representation** and are valid input to this function:

- **cartesian_point**;
- **degenerate_pcurve**;
- **point_on_curve**;
- **point_on_surface**.

The three latter ones in the list reference either **curves** or **surfaces** or both.

The **basis_surface** of a **degenerate_pcurve** may be any of the valid **surfaces** in a **geometrically_bounded_surface_shape_representation**. The **reference_to_curve** of a **degenerate_pcurve** shall be of one of the following types:

- **b_spline_curve**;
- **composite_curve** (recursive);
- **conic**;
- **curve_replica** (recursive);
- **line**;
- **polyline**;
- **trimmed_curve** (recursive).

The **basis_curve** of a **point_on_curve** may be any of the valid **curves** in a **geometrically_bounded_surface_shape_representation**.

The **basis_surface** of a **point_on_surface** may be any of the valid **surfaces** in a **geometrically_bounded_surface_shape_representation**.

This function returns TRUE, if the types of all referenced geometries are within the scope of the **geometrically_bounded_surface_shape_representation**, otherwise the function returns FALSE.

NOTE 1 This function does not check the correctness of references with respect to ISO 10303-42. Only additional requirements due to the scope of the **geometrically_bounded_surface_shape_representation** are checked.

NOTE 2 The function uses recursion to trace entity references.

EXPRESS specification:

```

*)
FUNCTION gbsf_check_point (pnt : point) : BOOLEAN;

(* This function varifies the validity of a point in the context of a
   geometrically bounded surface model.
*)
(* a cartesian_point is valid
*)
IF 'GEOMETRY_SCHEMA.CARTESIAN_POINT' IN TYPEOF (pnt) THEN
    RETURN(TRUE);
ELSE

(* a point_on_curve shall reference a valid curve
*)
IF 'GEOMETRY_SCHEMA.POINT_ON_CURVE' IN TYPEOF (pnt) THEN
    RETURN (gbsf_check_curve
            (pnt\point_on_curve.basis_curve));
ELSE

(* a point_on_surface shall reference a valid surface
*)
IF 'GEOMETRY_SCHEMA.POINT_ON_SURFACE' IN TYPEOF (pnt) THEN
    RETURN(gbsf_check_surface
            (pnt\point_on_surface.basis_surface));
ELSE

(* a degenerate_pcurve shall reference a valid curve and
   a valid surface
*)
IF 'GEOMETRY_SCHEMA.DEGENERATE_PCURVE' IN TYPEOF (pnt) THEN
    RETURN
        ((gbsf_check_curve
          (pnt\degenerate_pcurve.reference_to_curve\representation.items[1]))
         AND (gbsf_check_surface (pnt\degenerate_pcurve.basis_surface)));
    END_IF;
    END_IF;
    END_IF;
    END_IF;
    RETURN(FALSE);
END_FUNCTION;
(*

```

Argument definitions:

pnt: (input) the **point** that is being checked for being a valid **point** in a **geometrically_bounded_-surface_shape_representation**.

BOOLEAN: (return) is TRUE if the **point** is a valid **point**; else FALSE.

F.1.2.3.2 gbsf_check_curve

The **gbsf_check_curve** function performs type and constraint checking for **curves**. An instance is checked against all types of **curves** that are valid in the context of a **geometrically_bounded_surface_shape_representation**. All related geometry, that is **curves** and **surfaces**, are also checked in the same way.

EXAMPLE 1 One of the constraints that is validated by this function is whether the self-intersection flag of a **b_spline_curve** instance is set to TRUE or FALSE; only FALSE is valid.

Where appropriate an instance is investigated recursively. This means if a **curve** references another **curve** as a basis curve or parent curve, the **gbsf_check_curve** function is called again. If a **surface** is referenced, the **gbsf_check_surface** function is called. The recursion terminates at entity types that do not reference any **curves** or **surfaces**.

The following curve types and their subtypes are within the scope of the **geometrically_bounded_surface_shape_representation** and are valid input to this function:

- **b_spline_curve**;
- **composite_curve**;
- **conic**;
- **curve_replica**;
- **line**;
- **offset_curve_3d**;
- **pcurve**;
- **polyline**;
- **surface_curve**;
- **trimmed_curve**.

The **parent_curve** of a **curve_replica**, the **basis_curve** of an **offset_curve_3d**, and the **basis_curve** of a **trimmed_curve** shall all be of one of the following types:

- **b_spline_curve**;
- **composite_curve** (recursive);
- **conic**;

- **curve_replica** (recursive);
- **line**;
- **offset_curve_3d** (recursive);
- **pcurve** (recursive);
- **polyline**;
- **surface_curve** (recursive);
- **trimmed_curve** (recursive).

The one instance in the set of **items** of a **definitional_representation** that is referenced as **reference_to_curve** by a **pcurve** shall be of one of the following types:

- **b_spline_curve**;
- **composite_curve** (recursive);
- **conic**;
- **curve_replica** (recursive);
- **line**;
- **polyline**;
- **trimmed_curve** (recursive).

The **curve_3d** of a **surface_curve** shall be of one of the following types:

- **b_spline_curve**;
- **composite_curve** (recursive);
- **conic**;
- **curve_replica** (recursive);
- **line**;
- **offset_curve_3d** (recursive);
- **polyline**;

- **surface_curve** (recursive);
- **trimmed_curve** (recursive).

This function returns TRUE, if the types of all referenced geometries are within the scope of the **geometrically_bounded_surface_shape_representation** and if all constraints are satisfied, otherwise the function returns FALSE.

NOTE 1 This function does not check the correctness of references with respect to ISO 10303-42. ISO 10303-42 requires, for example, that the **segments** of a **composite_curve** shall be of type **composite_curve_segment** and that the **parent_curve** of a **composite_curve_segment** shall be a **bounded_curve**. This constraint is not verified by this function. Only additional requirements due to the scope of the **geometrically_bounded_surface_shape_representation** are checked.

NOTE 2 This function uses recursion to trace entity references.

EXPRESS specification:

*)

```
FUNCTION gbsf_check_curve (cv : curve) : BOOLEAN;
```

```
(* This function varifies the validity of a curve in the context of a
   geometrically bounded surface model.
```

```
*)
```

```
(* complex subtypes of curve that are both bounded_curve and one of
   conic, curve_replica, line, or offset_curve_3d are not valid
```

```
*)
```

```
IF SIZEOF ([ 'GEOMETRY_SCHEMA.BOUNDED_CURVE', 'GEOMETRY_SCHEMA.CONIC',
  'GEOMETRY_SCHEMA.CURVE_REPLICA', 'GEOMETRY_SCHEMA.LINE',
  'GEOMETRY_SCHEMA.OFFSET_CURVE_3D' ] * TYPEOF(cv)) > 1
```

```
THEN RETURN(FALSE);
```

```
ELSE
```

```
(* let those types pass that do not have any further references
   respectively rules to be applied
```

```
*)
```

```
IF SIZEOF ([ 'GEOMETRY_SCHEMA.CIRCLE', 'GEOMETRY_SCHEMA.ELLIPSE' ]
  * TYPEOF(cv)) = 1
```

```
THEN RETURN(TRUE);
```

```
ELSE
```

```
(* b_spline_curves shall not self-intersect
```

```
*)
```

```
IF (('GEOMETRY_SCHEMA.B_SPLINE_CURVE' IN TYPEOF(cv)) AND
  (cv.b_spline_curve.self_intersect = FALSE))
```

```
THEN RETURN(TRUE);
```

```
ELSE
```



```

(* a composite_curve shall not self-intersect and all of
   its segments shall reference valid curves
*)
IF (('GEOMETRY_SCHEMA.COMPOSITE_CURVE' IN TYPEOF(cv)) AND
    (cv\composite_curve.self_intersect = FALSE)) THEN
    RETURN (SIZEOF (QUERY (seg <* cv\composite_curve.segments |
        NOT (gbsf_check_curve(seg.parent_curve)))) = 0);
ELSE

    (* a curve_replica shall reference a valid curve
    *)
    IF 'GEOMETRY_SCHEMA.CURVE_REPLICA' IN TYPEOF(cv) THEN
        RETURN (gbsf_check_curve (cv\curve_replica.parent_curve));
    ELSE

        (* an offset_curve_3d shall not self-intersect and shall
           reference a valid curve
        *)
        IF (('GEOMETRY_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF(cv)) AND
            (cv\offset_curve_3d.self_intersect = FALSE)) THEN
            RETURN (gbsf_check_curve (cv\offset_curve_3d.basis_curve));
        ELSE

            (* a pcurve shall reference a valid curve and a valid
               basis_surface
            *)
            IF 'GEOMETRY_SCHEMA.PCURVE' IN TYPEOF(cv) THEN
                RETURN ((gbsf_check_curve
                    (cv\pcurve.reference_to_curve\representation.items[1])) AND
                    (gbsf_check_surface (cv\pcurve.basis_surface)));
            ELSE

                (* a polyline shall have at least 3 points and shall only
                   be used to represent an intersection_curve
                *)
                IF 'GEOMETRY_SCHEMA.POLYLINE' IN TYPEOF(cv) THEN
                    IF (SIZEOF (cv\polyline.points) >= 3) AND
                        (SIZEOF ((bag_to_set (USEDIN (cv, '')) -
                            bag_to_set (USEDIN (cv,
                                'GEOMETRY_SCHEMA.INTERSECTION_CURVE.CURVE_3D')))) = 0)
                        THEN RETURN (TRUE);
                    END_IF;
                ELSE

                    (* a surface_curve references a curve_3d and one or two
                       pcurves or one or two surfaces or one of each;
                       all of these references shall be valid
                    *)
                    IF 'GEOMETRY_SCHEMA.SURFACE_CURVE' IN TYPEOF(cv) THEN

                        (* if the curve reference is correct, check also the rest
                        *)

```

```

IF gbsf_check_curve (cv\surface_curve.curve_3d) THEN
  REPEAT i := 1 TO SIZEOF
    (cv\surface_curve.associated_geometry);

    (* do for one or two associated_geometrys:
    *)
  IF 'GEOMETRY_SCHEMA.SURFACE' IN TYPEOF
    (cv\surface_curve.associated_geometry[i]) THEN
    IF NOT gbsf_check_surface
      (cv\surface_curve.associated_geometry[i]) THEN
      RETURN(FALSE);
    END_IF;
  ELSE
    IF 'GEOMETRY_SCHEMA.PCURVE' IN TYPEOF
      (cv\surface_curve.associated_geometry[i]) THEN
      IF NOT gbsf_check_curve
        (cv\surface_curve.associated_geometry[i]) THEN
      RETURN(FALSE);
      END_IF;
    END_IF;
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_IF;
ELSE
  (* a trimmed_curve shall reference a valid
  basis_curve;
  line, parabola or hyperbola are only valid
  when referenced by a trimmed_curve
  *)
  IF 'GEOMETRY_SCHEMA.TRIMMED_CURVE' IN TYPEOF (cv) THEN

    (* line, parabola or hyperbola are valid without
    further checks
    *)
    IF SIZEOF (['GEOMETRY_SCHEMA.LINE',
      'GEOMETRY_SCHEMA.PARABOLA',
      'GEOMETRY_SCHEMA.HYPERBOLA'] *
      TYPEOF (cv\trimmed_curve.basis_curve)) = 1
    THEN RETURN (TRUE);

    (* check validity for all other curves
    *)
    ELSE RETURN (gbsf_check_curve
      (cv\trimmed_curve.basis_curve));
    END_IF;
  END_IF;
END_IF;
END_IF;
END_IF;
END_IF;
END_IF;

```

```

        END_IF;
    END_IF;
END_IF;
END_IF;
END_IF;
RETURN(FALSE);
END_FUNCTION;
( *

```

Argument definitions:

cv: (input) the **curve** that is being checked for a valid **curve** in a **geometrically_bounded_surface_shape_representation**.

BOOLEAN: (return) is TRUE if the **curve** is a valid **curve**; else FALSE.

F.1.2.3.3 gbsf_check_surface

The **gbsf_check_surface** function performs type and constraint checking for **surfaces**. An instance is checked against all types of **surfaces** that are valid in the context of a **geometrically_bounded_surface_shape_representation**. All related geometry, that is **curves** and **surfaces**, are also checked in the same way.

EXAMPLE 1 Whether the instance of a **b_spline_surface** self-intersects, is an example of a constraint that is validated by this function.

Where appropriate an instance is investigated recursively. This means if a **surface** references another **surface** as a basis surface or parent surface, the **gbsf_check_surface** function is called again. If a **curve** is referenced, the **gbsf_check_curve** function is called. The recursion terminates at entity types that do not reference any **curves** or **surfaces**.

The following surface types and their subtypes are within the scope of the **geometrically_bounded_surface_shape_representation** and are valid input to this function:

- **b_spline_surface;**
- **curve_bounded_surface;**
- **elementary_surface;**
- **offset_surface;**
- **rectangular_composite_surface;**
- **rectangular_trimmed_surface;**
- **surface_replica;**

– **swept_surface**.

Both the **basis_surface** of a **curve_bounded_surface**, the **basis_surface** of an **offset_surface**, the **basis_surface** of a **rectangular_trimmed_surface**, and the **parent_surface** of a **surface_replica** shall all be one of the **surface** types listed above.

Swept_surfaces reference **curves**. Function **gbsf_check_curve** is called for type checking. The **geometrically_bounded_surface_shape_representation** requires the same constraints on valid sweeping curves as specified in ISO 10303-42. All **curves** that are in the scope of a **geometrically_bounded_surface_shape_representation** are valid **swept_curves**.

This function returns TRUE, if the types of all referenced geometries are within the scope of the **geometrically_bounded_surface_shape_representation** and if all constraints are satisfied, otherwise the function returns FALSE.

NOTE 1 This function does not check the correctness of references with respect to ISO 10303-42. ISO 10303-42 requires, for example, that the **segments** of a **rectangular_composite_surface** shall be **surface_patches** and that these shall have **bounded_surfaces** as their parent surfaces. This constraint is not verified by this function. Only additional requirements due to the scope of the **geometrically_bounded_surface_shape_representation** are checked.

NOTE 2 The function uses recursion to trace entity references.

EXPRESS specification:

*)

FUNCTION gbsf_check_surface (sf : surface) : BOOLEAN;

(* This function verifies the validity of a surface in the context of a geometrically bounded surface model.

*)

(* a b_spline_surface shall not self-intersect

*)

IF (('GEOMETRY_SCHEMA.B_SPLINE_SURFACE' IN TYPEOF(sf)) AND
(sf\b_spline_surface.self_intersect = FALSE)) THEN
RETURN(TRUE);

ELSE

(* spherical_surface and toroidal_surface are valid

*)

IF SIZEOF (['GEOMETRY_SCHEMA.SPHERICAL_SURFACE',
'GEOMETRY_SCHEMA.TOROIDAL_SURFACE'] * TYPEOF(sf)) = 1 THEN
RETURN(TRUE);

ELSE

(* a curve_bounded_surface shall reference a valid surface
and valid curves

```

*)
IF 'GEOMETRY_SCHEMA.CURVE_BOUNDED_SURFACE' IN TYPEOF(sf) THEN

  (* if there is a simple basis_surface, check the curves
  *)
  IF SIZEOF (['GEOMETRY_SCHEMA.CONICAL_SURFACE',
    'GEOMETRY_SCHEMA.CYLINDRICAL_SURFACE',
    'GEOMETRY_SCHEMA.PLANE'])
    * TYPEOF(sf\curve_bounded_surface.basis_surface)) = 1 THEN
    RETURN(SIZEOF (QUERY (bcurve <*
      sf\curve_bounded_surface.boundaries |
      NOT (gbsf_check_curve(bcurve)))) = 0);
  ELSE

    (* for all other surfaces check recursively the
    basis_surface and then the curves
    *)
    IF gbsf_check_surface
      (sf\curve_bounded_surface.basis_surface) THEN
      RETURN(SIZEOF (QUERY (bcurve <*
        sf\curve_bounded_surface.boundaries |
        NOT (gbsf_check_curve(bcurve)))) = 0);
    END_IF;
  END_IF;
ELSE

  (* an offset_surface shall not self-intersect and shall
  reference a valid surface
  *)
  IF (('GEOMETRY_SCHEMA.OFFSET_SURFACE' IN TYPEOF(sf)) AND
    (sf\offset_surface.self_intersect = FALSE)) THEN
    RETURN (gbsf_check_surface (sf\offset_surface.basis_surface));
  ELSE

    (* a rectangular_composite_surface shall reference a matrix of
    valid surfaces
    *)
    IF 'GEOMETRY_SCHEMA.RECTANGULAR_COMPOSITE_SURFACE' IN TYPEOF(sf)
    THEN
      REPEAT i := 1 TO SIZEOF
        (sf\rectangular_composite_surface.segments);
      REPEAT j := 1 TO SIZEOF
        (sf\rectangular_composite_surface.segments[i]);
      IF NOT (gbsf_check_surface
        (sf\rectangular_composite_surface.segments[i][j].
        parent_surface))
      THEN RETURN (FALSE);
      END_IF;
    END_REPEAT;
    END_REPEAT;
    RETURN(TRUE);
  ELSE

```

```

(* a rectangular_trimmed_surface shall have a valid
   basis_surface
*)
IF 'GEOMETRY_SCHEMA.RECTANGULAR_TRIMMED_SURFACE' IN TYPEOF(sf)
THEN
    IF  SIZEOF (['GEOMETRY_SCHEMA.CONICAL_SURFACE',
                 'GEOMETRY_SCHEMA.CYLINDRICAL_SURFACE',
                 'GEOMETRY_SCHEMA.PLANE'] * TYPEOF
               (sf\rectangular_trimmed_surface.basis_surface)) = 1 THEN
        RETURN (TRUE);
    ELSE
        RETURN(gbsf_check_surface
               (sf\rectangular_trimmed_surface.basis_surface));
    END_IF;
ELSE

    (* a surface_replica shall have a valid parent surface
       *)
    IF 'GEOMETRY_SCHEMA.SURFACE_REPLICA' IN TYPEOF (sf) THEN
        RETURN (gbsf_check_surface
                (sf\surface_replica.parent_surface));
    ELSE

        (* a swept_surface shall have a valid sweeping curve
           *)
        IF 'GEOMETRY_SCHEMA.SWEPT_SURFACE' IN TYPEOF(sf) THEN
            RETURN(gbsf_check_curve (sf\swept_surface.swept_curve));
        END_IF;
    END_IF;
END_IF;
END_IF;
END_IF;
END_IF;
END_IF;
END_IF;
RETURN(FALSE);
END_FUNCTION;
(*

```

Argument definitions:

sf: (input) the **surface** that is being checked for a valid **surface** in a **geometrically_bounded_surface_shape_representation**.

BOOLEAN: (return) is TRUE if the **surface** is a valid **surface**; else FALSE.

F.1.2.3.4 bag_to_set

This function converts BAGs into SETs.

EXAMPLE 1 It can be used to convert the BAGs returned by the USEDIN function into SETs that can be properly assigned to variables that are SETs.

NOTE 1 This function specification is copied from ISO 10303-41. It is included in this part of ISO 10303 to resolve all references from this AIC schema. The function is unchanged.

EXPRESS specification:

```

*)
FUNCTION bag_to_set (the_bag : BAG OF GENERIC : intype) :
    SET OF GENERIC : intype;

    LOCAL
        the_set: SET OF GENERIC : intype := [];
        i      : INTEGER;
    END_LOCAL;

    IF SIZEOF (the_bag) > 0 THEN
        REPEAT i := 1 to HIINDEX (the_bag);
            the_set := the_set + the_bag [i];
        END_REPEAT;
    END_IF;

    RETURN (the_set);

END_FUNCTION;
( *

```

Argument definitions:

the_bag: the BAG that is to be converted into a SET.

```

*)
END_SCHEMA; -- aic_geometrically_bounded_surface
( *

```

F.2 Manifold surface

F.2.1 Scope

This application interpreted construct specifies the interpretation of the integrated resources to satisfy requirements for the description of geometric shapes by means of manifold surface models.

The following are within the scope of this AIC:

— 3D points;

- points defined in the parameter space of curves or surfaces;
- 3D curves;
- curves defined in the parameter space of surfaces;

NOTE Such curves are also known as pcurves or cons , which are acronyms for parametrised curve and curve on surface .

- the elementary curves, i.e., line, circle, ellipse, parabola, and hyperbola;
- intersection curves;
- polylines that consist of at least three points;
- the elementary surfaces, i.e., plane, cylinder, cone, torus, and sphere;
- swept surfaces created by rotation or linear extrusion of a curve;
- sculptured curves and surfaces;
- trimming of curves and surfaces using topological entities;
- composition of curves and surfaces using topological entities;
- replication of curves, surfaces, and surface models;
- 3D offsets of curves and surfaces;
- 2-manifolds.

The following are outside the scope of this AIC:

- unbounded geometry;
- self-intersecting geometry;
- geometry in a 2D cartesian coordinate space;
- replication of points;
- topology without an association to a corresponding geometric domain;
- non-manifolds.

F.2.2 EXPRESS short listing

This application interpreted construct provides a consistent set of geometric and topological entities for the definition of manifold surface representations that consist of elementary or sculptured curves and surfaces. The highest level entity of this AIC is **manifold_surface_shape_representation**. A **manifold_surface_shape_representation** is bounded. The bounding of the geometry is achieved by topological entities, such as **vertex**, **edge**, and **face**.

Topological entities shall not exist without an association to a corresponding geometric domain.

NOTE 1 This AIC uses all the entities and types from ISO 10303-511, **aic_topologically_bounded_surface**.

EXPRESS specification:

```

*)
SCHEMA aic_manifold_surface;

    USE FROM aic_topologically_bounded_surface;

    USE FROM geometric_model_schema (
        shell_based_surface_model);

    USE FROM geometry_schema (
        b_spline_curve,
        b_spline_surface,
        bounded_pcurve,
        bounded_surface_curve,
        cartesian_transformation_operator_3d,
        curve,
        curve_replica,
        degenerate_pcurve,
        evaluated_degenerate_pcurve,
        intersection_curve,
        offset_curve_3d,
        offset_surface,
        point_on_curve,
        point_on_surface,
        seam_curve,
        surface,
        surface_replica);

    USE FROM product_property_representation_schema (
        shape_representation);

    USE FROM representation_schema (
        mapped_item,
        representation,
        representation_map);

```

```
USE FROM topology_schema (
    closed_shell,
    connected_face_set,
    face,
    open_shell,
    oriented_closed_shell,
    oriented_face);
```

```
( *
```

NOTE 2 The schemas referenced above can be found in the following parts of ISO 10303:

aic_topologically_bounded_surface	ISO 10303-511
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
product_property_representation_schema	ISO 10303-41
representation_schema	ISO 10303-43
topology_schema	ISO 10303-42

F.2.2.1 aic_manifold_surface entity definition: manifold_surface_shape_representation

A **manifold_surface_representation** describes the shape or portions of the shape of a **product** using 2-manifolds with boundaries.

NOTE 1 2-manifolds are topologically constrained in a way that makes them suitable for inclusion into solid representations. Within a 2-manifold, for example, only a maximum of two **faces** may share the same **edge**.

NOTE 2 Entity **product** is not part of this AIC.

A **manifold_surface_shape_representation** is a **shape_representation** (see ISO 10303-41) that consists of one or many **shell_based_surface_models**. Each **shell_based_surface_model** is built up of **open_shells** and **closed_shells** which are sets of **faces**. **Connected_face_set**, which is the supertype of **open_shell** and **closed_shell**, shall not be instantiated. **Faces** use **edges** and **vertices**; all of these shall reference geometric entities such as **points**, **curves**, and **surfaces**. The link between topology and geometry may be established by either using **face_surface** or **advanced_face** (see ISO 10303-511). The two options differ both in the selection of valid **point**, **curve**, and **surface** subtypes and in constraints on references to underlying geometry.

NOTE 3 The representation of **face_surfaces** as **advanced_faces** is recommended for manifold surface models that are intended to be used together with boundary representation solid models; the integration of such a manifold surface model into for example an advanced boundary representation model (see ISO 10303-514) will be easier.

All geometric entities shall be of dimensionality three except for two-dimensional geometry that is used for the purpose of defining **pcurves**. The use of one-dimensional **cartesian_points** is excluded.

All unbounded geometry shall be trimmed by using topological constructs.

The **items** of a **manifold_surface_shape_representation** may also be of type **mapped_item** (see ISO 10303-43) or **axis2_placement_3d**. These are used to assemble one or several **manifold_surface_shape_representations** into one other **manifold_surface_shape_representation**.

The WHERE-rules of this entity restrict the use of the entity data types that are imported from ISO 10303-42 and ISO 10303-43 according to the statements above. Some of these type checks and validations of constraints are specified in the following two functions:

- **msf_curve_check**;
- **msf_surface_check**.

In WR6 and WR11 below, these functions validate the **curves** and **surfaces** of all **edges** and **faces** that are in the scope of a **manifold_surface_shape_representation** except for those that are in the reference tree of an **advanced_face**; the geometry of an **advanced_face** is validated by a different set of rules. The functions automatically assess all underlying geometry; this is done by recursive calls.

EXAMPLE 1 A **pcurve** may reference both a **curve** and a **surface**. Function **msf_curve_check** validates not only the **pcurve**, but also this underlying geometry. It will, therefore, not only call itself, but also **msf_surface_check**.

NOTE 4 This part of ISO 10303 does not include a function for the validation of **points** and their underlying **curves** and **surfaces**. This is because all **curves** and **surfaces** of a **manifold_surface_shape_representation** are referenced from **edges** and **faces** and are, thus, covered by the two existing functions already.

NOTE 5 An application protocol that uses this AIC should explicitly require that the **shape_representation** entity may be instantiated as a **manifold_surface_shape_representation**.

EXPRESS specification:

```

*)
ENTITY manifold_surface_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
    NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL',
      'AIC_MANIFOLD_SURFACE.MAPPED_ITEM',
      'AIC_MANIFOLD_SURFACE.AXIS2_PLACEMENT_3D'] * TYPEOF (it)) = 1))) = 0;
  WR2: SIZEOF (QUERY (it <* SELF.items |
    SIZEOF ([ 'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL',
      'AIC_MANIFOLD_SURFACE.MAPPED_ITEM'] * TYPEOF (it)) = 1)) > 0;
  WR3: SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
    'AIC_MANIFOLD_SURFACE.MAPPED_ITEM' IN TYPEOF (it)) |
    NOT ('AIC_MANIFOLD_SURFACE.MANIFOLD_SURFACE_SHAPE_REPRESENTATION'
    IN TYPEOF (mi\mapped_item.mapping_source.mapped_representation))))
    = 0;
  WR4: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |

```

```

'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (sh <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.OPEN_SHELL',
'AIC_MANIFOLD_SURFACE.ORIENTED_CLOSED_SHELL',
'AIC_MANIFOLD_SURFACE.CLOSED_SHELL']
* TYPEOF (sh)) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.FACE_SURFACE',
'AIC_MANIFOLD_SURFACE.ORIENTED_FACE'] * TYPEOF (fa)) = 1))) = 0)))
= 0))) = 0;
WR6: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (f_sf <* QUERY (fa <*
cfs\connected_face_set.cfs_faces |
'AIC_MANIFOLD_SURFACE.FACE_SURFACE' IN TYPEOF (fa)) |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (f_sf))
OR
(SIZEOF ([ 'AIC_MANIFOLD_SURFACE.B_SPLINE_SURFACE',
'AIC_MANIFOLD_SURFACE.ELEMENTARY_SURFACE',
'AIC_MANIFOLD_SURFACE.OFFSET_SURFACE',
'AIC_MANIFOLD_SURFACE.SURFACE_REPLICA',
'AIC_MANIFOLD_SURFACE.SWEPT_SURFACE'] * TYPEOF
(f_sf\face_surface.face_geometry)) = 1)))) = 0))) = 0))) = 0;
WR7: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(msf_surface_check(fa\face_surface.face_geometry)))))) = 0)))
= 0))) = 0;
WR8: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (bnds <* fa.bounds |
NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.EDGE_LOOP',
'AIC_MANIFOLD_SURFACE.VERTEX_LOOP']
* TYPEOF (bnds.bound)) = 1))) = 0))) = 0))) = 0))) = 0;
WR9: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |

```

```

NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.EDGE_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
NOT ('AIC_MANIFOLD_SURFACE.EDGE_CURVE' IN TYPEOF
(oe.edge_element)))) = 0))) = 0))) = 0))) = 0;
WR10: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.EDGE_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe_cv <* QUERY (oe <*
elp_fbnds\path.edge_list |
'AIC_MANIFOLD_SURFACE.EDGE_CURVE' IN TYPEOF (oe.edge_element)) |
NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.B_SPLINE_CURVE',
'AIC_MANIFOLD_SURFACE.CONIC',
'AIC_MANIFOLD_SURFACE.CURVE_REPLICA',
'AIC_MANIFOLD_SURFACE.LINE',
'AIC_MANIFOLD_SURFACE.OFFSET_CURVE_3D',
'AIC_MANIFOLD_SURFACE.PCURVE',
'AIC_MANIFOLD_SURFACE.POLYLINE',
'AIC_MANIFOLD_SURFACE.SURFACE_CURVE'] *
TYPEOF (oe_cv.edge_element\edge_curve.edge_geometry))
= 1))) = 0))) = 0))) = 0))) = 0;
WR11: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.EDGE_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
NOT (msf_curve_check (oe.edge_element\edge_curve.edge_geometry))))
= 0))) = 0))) = 0))) = 0;
WR12: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |

```

```

        'AIC_MANIFOLD_SURFACE.EDGE_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
NOT (('AIC_MANIFOLD_SURFACE.VERTEX_POINT' IN TYPEOF
(oe.edge_element.edge_start))
AND
('AIC_MANIFOLD_SURFACE.VERTEX_POINT' IN
TYPEOF (oe.edge_element.edge_end))))))
= 0))) = 0))) = 0))) = 0))) = 0;
WR13: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.EDGE_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
NOT ((SIZEOF ([ 'AIC_MANIFOLD_SURFACE.CARTESIAN_POINT',
'AIC_MANIFOLD_SURFACE.DEGENERATE_PCURVE',
'AIC_MANIFOLD_SURFACE.POINT_ON_CURVE',
'AIC_MANIFOLD_SURFACE.POINT_ON_SURFACE'] * TYPEOF
(oe.edge_element.edge_start\vertex_point.vertex_geometry)) = 1)
AND
(SIZEOF ([ 'AIC_MANIFOLD_SURFACE.CARTESIAN_POINT',
'AIC_MANIFOLD_SURFACE.DEGENERATE_PCURVE',
'AIC_MANIFOLD_SURFACE.POINT_ON_CURVE',
'AIC_MANIFOLD_SURFACE.POINT_ON_SURFACE'] * TYPEOF
(oe.edge_element.edge_end\vertex_point.vertex_geometry)) = 1
)))) = 0))) = 0))) = 0))) = 0))) = 0;
WR14: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.VERTEX_LOOP' IN TYPEOF (bnds.bound)) |
NOT ('AIC_MANIFOLD_SURFACE.VERTEX_POINT' IN TYPEOF
(vlp_fbnds\vertex_loop.loop_vertex)))) = 0))) = 0)))
= 0))) = 0;
WR15: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
'AIC_MANIFOLD_SURFACE.SHELL_BASED_SURFACE_MODEL' IN TYPEOF (it)) |
NOT (SIZEOF (QUERY (cfs <*
sbsm\shell_based_surface_model.sbsm_boundary |
NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
NOT (('AIC_MANIFOLD_SURFACE.ADVANCED_FACE' IN TYPEOF (fa))
OR
(SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* fa.bounds |
'AIC_MANIFOLD_SURFACE.VERTEX_LOOP' IN TYPEOF (bnds.bound)) |
NOT (SIZEOF ([ 'AIC_MANIFOLD_SURFACE.CARTESIAN_POINT',

```

```

        'AIC_MANIFOLD_SURFACE.DEGENERATE_PCURVE',
        'AIC_MANIFOLD_SURFACE.POINT_ON_CURVE',
        'AIC_MANIFOLD_SURFACE.POINT_ON_SURFACE'] * TYPEOF
        (vlp_fbnds\vertex_loop.loop_vertex\vertex_point.vertex_geometry))
        = 1))) = 0)))) = 0))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **items** in a **manifold_surface_shape_representation** shall be **shell_based_surface_models**, **mapped_items**, or **axis2_placement_3ds**.

NOTE 6 **Axis2_placement_3d** is a valid **mapped_item.mapping_target**. To include another **representation** into the list of **items** of a **manifold_surface_shape_representation** (see WR3 for valid **mapped_items**), the **mapped_item.mapping_source.mapping_origin** may be any entity that is geometrically founded in the **geometric_representation_context** of the **mapped_representation**. If this entity is an **axis2_placement_3d**, the operator that maps the **mapped_representation** into the **manifold_surface_shape_representation** corresponds to a transformation matrix with only translation and rotation enabled. If a **cartesian_transformation_operator_3d** is used as **mapping_origin**, scaling and mirroring are possible.

WR2: At least one of the **items** in a **manifold_surface_shape_representation** shall be either a **shell_based_surface_model** or a **mapped_item**.

WR3: If there is a **mapped_item** in a **manifold_surface_shape_representation**, the **mapped_representation** of its **mapping_source** shall also be a **manifold_surface_shape_representation**.

WR4: The boundary of a **shell_based_surface_model**, that is its attribute **sbsm_boundary**, shall be defined by using only **open_shells**, **closed_shells**, or **oriented_closed_shells**; **oriented_open_shells** are not valid.

WR5: A **face** shall be instantiated as either a **face_surface**, its subtype **advanced_face**, or an **oriented_face**.

WR6: The **face_geometry** of a **face_surface** shall be either a **b_spline_surface**, an **elementary_surface**, an **offset_surface**, a **surface_replica**, a **swept_surface**, or a **surface** within the reference tree of an **advanced_face**.

WR7: All basis geometry that is referenced by **surfaces** shall be either within the reference tree of an **advanced_face** or shall be valid **curves** and **surfaces**.

The **basis_surface** of an **offset_surface** shall be either an **elementary_surface**, **b_spline_surface**, **offset_surface**, **swept_surface**, or **surface_replica**.

The **parent_surface** of a **surface_replica** shall be either an **elementary_surface**, **b_spline_surface**, **offset_surface**, **swept_surface**, or **surface_replica**.

The **swept_curve** of a **swept_surface** shall be either a **line**, **conic**, **pcurve**, **surface_curve**, **offset_curve_3d**, **b_spline_curve**, **polyline**, or **curve_replica**.

The attribute **self_intersect** shall be set to FALSE for **b_spline_surfaces** and **offset_surfaces**.

WR8: The **bound** of a **face_bound**, which is referenced from a **face**, shall be either within the reference tree of an **advanced_face** or shall be an **edge_loop** or a **vertex_loop**.

WR9: The geometry of a bounding **edge**, that is the **edge_element** of an **oriented_edge**, shall be either within the reference tree of an **advanced_face** or shall be an **edge_curve**.

WR10: The **curve** used to define the geometry of an **edge**, that is the **edge_geometry** of an **edge_curve** of an **edge**, shall be either a **b_spline_curve**, a **conic**, a **curve_replica**, a **line**, an **offset_curve_3d**, a **pcurve**, a **polyline**, a **surface_curve**, or a **curve** in the reference tree of an **advanced_face**.

WR11: All basis geometry that is referenced by **curves** shall be either within the reference tree of an **advanced_face** or shall be valid **curves** and **surfaces**.

The **parent_curve** of a **curve_replica** shall be either a **line**, **conic**, **pcurve**, **surface_curve**, **offset_curve_3d**, **b_spline_curve**, **polyline**, or **curve_replica**.

The **basis_curve** of an **offset_curve_3d** shall be either a **line**, **conic**, **pcurve**, **surface_curve**, **offset_curve_3d**, **b_spline_curve**, **polyline**, or **curve_replica**.

The **curve_3d** of a **surface_curve** shall be either a **line**, **conic**, **offset_curve_3d**, **b_spline_curve**, **polyline**, or **curve_replica**.

The **basis_surface** of a **surface_curve** shall be either a **b_spline_surface**, **elementary_surface**, **offset_surface**, **surface_replica**, or **swept_surface**.

Polylines shall contain at least three **cartesian_points**.

The attribute **self_intersect** shall be set to FALSE for **b_spline_curves** and **offset_curve_3ds**.

WR12: The **edge_start** and **edge_end** of an **edge** shall be either within the reference tree of an **advanced_face** or shall be **vertex_points**.

WR13: The **vertex_geometry** of a **vertex** that is part of an **edge_loop** shall be either within the reference tree of an **advanced_face** or shall be a **cartesian_point**, **point_on_curve**, **point_on_surface**, or **degenerate_pcurve**.

WR14: The **loop_vertex** of a **vertex_loop** shall be either within the reference tree of an **advanced_face** or shall be a **vertex_point**.

WR15: The **vertex_geometry** of a **vertex** that is part of a **vertex_loop** shall be either within the reference tree of an **advanced_face** or shall be a **cartesian_point**, **point_on_curve**, **point_on_surface**, or **degenerate_pcurve**.

Informal propositions:

IP1: A **b_spline_curve** shall not self-intersect.

IP2: A **b_spline_surface** shall not self-intersect.

IP3: An **offset_curve_3d** shall not self-intersect.

IP4: An **offset_surface** shall not self-intersect.

IP5: If a **face** has only one connected outer bound, the corresponding **loop** shall be represented as **face_outer_bound**. If the outer bound is not connected, **face_outer_bound** shall not be used.

F.2.2.2 **aic_manifold_surface** function definitions

This section describes functions required to formulate constraints for the **aic_manifold_surface** schema. These functions are used in the specification of the entity **manifold_surface_shape_representation**.

F.2.2.2.1 **msf_curve_check**

The **msf_curve_check** function performs type checking for **curves**. An instance is checked against all types of **curves** that are valid in the context of a **manifold_surface_shape_representation**. All related geometry, that is **curves** and **surfaces**, are also checked in the same way.

Where appropriate an instance is investigated recursively. This means if a **curve** references another **curve** as a basis curve or parent curve, the **msf_curve_check** function is called again. If a **surface** is referenced, the **msf_surface_check** function is called. The recursion terminates at entity types that do not reference any **curves** or **surfaces**.

The following curve types and their subtypes are within the scope of the **manifold_surface_shape_representation** and are, thus, valid input to this function:

- **b_spline_curve**;
- **conic**;
- **curve_replica**;
- **line**;
- **offset_curve_3d**;
- **pcurve**;
- **polyline**;
- **surface_curve**.

Four of these **curve** types reference basis or parent **curves**. The lists below indicate the valid references.

The **parent_curve** of a **curve_replica** and the **basis_curve** of an **offset_curve_3d** shall be of one of the following types:

- **b_spline_curve**;
- **conic**;

- **curve_replica** (recursive);
- **line**;
- **offset_curve_3d** (recursive);
- **pcurve** (recursive);
- **polyline**;
- **surface_curve** (recursive).

The one instance in the set of **items** of a **definitional_representation** that is referenced as **reference_to_curve** by a **pcurve** shall be of one of the following types:

- **b_spline_curve**;
- **conic**;
- **curve_replica** (recursive);
- **line**;
- **polyline**.

The **curve_3d** of a **surface_curve** shall be of one of the following types:

- **b_spline_curve**;
- **conic**;
- **curve_replica** (recursive);
- **line**;
- **offset_curve_3d** (recursive);
- **polyline**;
- **surface_curve** (recursive).

Pcurve and **surface_curve** also reference **surfaces**. Function **msf_surface_check** is called for type checking. The **manifold_surface_shape_representation** requires the same constraints on valid **surface** references for **pcurves** and **surface_curves** as specified in ISO 10303-42.

A valid **polyline** shall consist of at least three **cartesian_points**. The attribute **self_intersect** shall for B-spline and offset geometry be set to FALSE.

This function returns TRUE, if the types of all referenced geometries are within the scope of the **manifold_surface_shape_representation** and if all constraints are fulfilled, otherwise the function returns FALSE.

NOTE 1 This function does not check the correctness of references with respect to ISO 10303-42. Only additional requirements due to the scope of the **manifold_surface_shape_representation** are checked.

NOTE 2 This function uses recursion to trace entity references.

EXPRESS specification:

```

*)
FUNCTION msf_curve_check (cv: curve) : BOOLEAN;

(* This function varifies the validity of a curve in the context
   of a manifold surface model.
*)

(* complex subtypes of curve that are both bounded_curve and
   oneof conic, curve_replica, line, or offset_curve_3d are not
   valid
*)
IF SIZEOF (['GEOMETRY_SCHEMA.BOUNDED_CURVE',
            'GEOMETRY_SCHEMA.CONIC',
            'GEOMETRY_SCHEMA.CURVE_REPLICA', 'GEOMETRY_SCHEMA.LINE',
            'GEOMETRY_SCHEMA.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1
THEN RETURN(FALSE);
ELSE

    (* b_spline_curves shall not self-intersect
    *)
    IF (('GEOMETRY_SCHEMA.B_SPLINE_CURVE' IN TYPEOF (cv)) AND
        (cv\b_spline_curve.self_intersect = FALSE))
    THEN RETURN(TRUE);
    ELSE

        (* let those types pass that do not have any further references
           respectively rules to be applied
        *)
        IF SIZEOF (['GEOMETRY_SCHEMA.CONIC', 'GEOMETRY_SCHEMA.LINE']
            * TYPEOF (cv)) = 1 THEN
            RETURN(TRUE);
        ELSE

            (* a curve_replica shall reference a valid curve
            *)
            IF 'GEOMETRY_SCHEMA.CURVE_REPLICA' IN TYPEOF(cv) THEN
                RETURN (msf_curve_check(cv\curve_replica.parent_curve));
            ELSE

```

```

(* an offset_curve_3d shall not self-intersect and
   shall reference a valid curve
*)
IF (('GEOMETRY_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF (cv)) AND
    (cv\offset_curve_3d.self_intersect = FALSE)) THEN
    RETURN (msf_curve_check(cv\offset_curve_3d.basis_curve));
ELSE

    (* a pcurve shall reference a valid curve and a valid
       basis_surface
    *)
    IF 'GEOMETRY_SCHEMA.PCURVE' IN TYPEOF(cv) THEN
        RETURN ((msf_curve_check
            (cv\pcurve.reference_to_curve\representation.items[1])) AND
            (msf_surface_check(cv\pcurve.basis_surface)));
    ELSE

        (* a surface_curve references a curve_3d and one or
           two pcurves or one or two surfaces or one of
           each; all of these references shall be valid
        *)
        IF 'GEOMETRY_SCHEMA.SURFACE_CURVE' IN TYPEOF(cv) THEN

            (* if the curve reference is correct, check also the rest
            *)
            IF msf_curve_check(cv\surface_curve.curve_3d) THEN
                REPEAT i := 1 TO SIZEOF
                    (cv\surface_curve.associated_geometry);

                    (* do for one or two associated_geometries:
                    *)
                    IF 'GEOMETRY_SCHEMA.SURFACE' IN
                        TYPEOF (cv\surface_curve.associated_geometry[i]) THEN
                        IF NOT msf_surface_check
                            (cv\surface_curve.associated_geometry[i]) THEN
                            RETURN(FALSE);
                        END_IF;
                    ELSE
                        IF 'GEOMETRY_SCHEMA.PCURVE' IN TYPEOF
                            (cv\surface_curve.associated_geometry[i]) THEN
                            IF NOT msf_curve_check
                                (cv\surface_curve.associated_geometry[i]) THEN
                                RETURN(FALSE);
                            END_IF;
                        END_IF;
                    END_IF;
                END_REPEAT;
                RETURN(TRUE);
            END_IF;
        ELSE

```

[illegible]

Argument definitions:

cv: (input) the **curve** that is being checked for a valid **curve** in a **manifold_surface_shape_representation**.

BOOLEAN: (output) is TRUE if the **curve** is a valid **curve**; else FALSE.

F.2.2.2.2 msf_surface_check

The **msf_surface_check** function performs type checking for **surfaces**. An instance is checked against all types of **surfaces** that are valid in the context of a **manifold_surface_shape_representation**. All related geometry, that is **curves** and **surfaces**, are also checked in the same way.

Where appropriate an instance is investigated recursively. This means if a **surface** references another **surface** as a basis surface or parent surface the **msf_surface_check** function is called again. If a **curve** is referenced, the **msf_curve_check** function is called. The recursion terminates at entity types that do not reference any **curves** or **surfaces**.

The following surface types and their subtypes are within the scope of the **manifold_surface_shape-representation** and are, thus, valid input to this function:

- `b_spline_surface`;
- `elementary_surface`;
- `offset_surface`;
- `surface_replica`;
- `swept_surface`.

Two of these **surface** types reference basis or parent **surfaces**. The **parent_surface** of a **surface_replica** and the **basis_surface** of an **offset_surface** shall both be of one of the following types:

- **b_spline_surface**;
- **elementary_surface**;
- **offset_surface** (recursive);
- **surface_replica** (recursive);
- **swept_surface**.

Swept_surfaces reference **curves**. Function **msf_curve_check** is called for type checking. The **manifold_surface_shape_representation** requires the same constraints on valid sweeping curves as specified in ISO 10303-42. All **curves** that are in the scope of a **manifold_surface_shape_representation** are valid as **swept_surface.swept_curves**.

The attribute **self_intersect** shall for B-spline and offset geometry be set to FALSE.

This function returns TRUE, if the types of all referenced geometries are within the scope of the **manifold_surface_shape_representation** and if all constraints are fulfilled, otherwise the function returns FALSE.

NOTE 1 This function does not check the correctness of references with respect to requirements specified by ISO 10303-42. Only additional requirements due to the scope of the **manifold_surface_shape_representation** are checked.

NOTE 2 The function uses recursion to trace entity references.

EXPRESS specification:

```

*)
FUNCTION msf_surface_check (surf : surface) : BOOLEAN;

(* This function verifies the validity of a surface in the
   context of a manifold surface model.
*)

(* let those types pass that do not have any further
   references respectively rules to be applied
   *)
IF 'GEOMETRY_SCHEMA.ELEMENTARY_SURFACE' IN TYPEOF(surf) THEN
  RETURN(TRUE);
ELSE

```

```

(* a swept_surface shall have a valid sweeping curve
*)
IF 'GEOMETRY_SCHEMA.SWEPT_SURFACE' IN TYPEOF (surf) THEN
    RETURN (msf_curve_check(surf\swept_surface.swept_curve));
ELSE

    (* an offset_surface shall not self-intersect and shall
       reference a valid surface
    *)
    IF (('GEOMETRY_SCHEMA.OFFSET_SURFACE' IN TYPEOF (surf)) AND
        (surf\offset_surface.self_intersect = FALSE)) THEN
        RETURN (msf_surface_check(surf\offset_surface.basis_surface));
    ELSE

        (* a surface_replica shall have a valid parent surface
        *)
        IF 'GEOMETRY_SCHEMA.SURFACE_REPLICA' IN TYPEOF(surf) THEN
            RETURN(msf_surface_check(surf\surface_replica.parent_surface));
        ELSE

            (* a b_spline_surface shall not self-intersect
            *)
            IF (('GEOMETRY_SCHEMA.B_SPLINE_SURFACE' IN TYPEOF(surf)) AND
                (surf\b_spline_surface.self_intersect = FALSE)) THEN
                RETURN(TRUE);
            END_IF;
        END_IF;
    END_IF;
END_IF;
RETURN(FALSE);
END_FUNCTION;
(*

```

Argument definitions:

surf: (input) the **surface** that is being checked for a valid **surface** in a **manifold_surface_shape_-representation**.

BOOLEAN: (output) is TRUE if the **surface** is a valid **surface**; else FALSE.

```

*)
END_SCHEMA; -- aic_manifold_surface
(*

```

Annex G (informative) **Application activity model**

The application activity model (AAM) is provided to aid in understanding the scope and information requirements defined in this application protocol. The model is presented as a set of figures that contain the activity diagrams and a set of definitions of the activities and their data. The application activity model is given in figures G.1 through G.21. The viewpoint of the application activity model is that of design participants in creating an electrical or electronic product.

G.1 Application activity model definitions

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol. The definitions given in this annex do not supersede the definitions given in the main body of the text.

G.1.1

advanced parts list

a preliminary list of all parts that go into the manufacture of a fully populated printed circuit board (see preliminary parts list.)

G.1.2

analyse board architecture manufacturability

an activity that reviews the architecture and ascertains the relative manufacturability of the product. Issues recorded are presented during meetings of the product development team.

G.1.3

analyse board architecture performance*

an activity that includes sub-activities that focus on the expected performance of the electrical or electronic product. Questions of acceptable performance are presented to the customer for final acceptance.

G.1.4

analyse board function*

an activity that includes sub-activities associated with verifying that the electrical or electronic product functionally meets the requirements of the customer. Results of these analyses are reported and discussed during project development team meetings.

G.1.5

analyse board manufacturability*

an activity that includes those sub-activities that address the ease with which the electrical or electronic product is manufacturable.

G.1.6**analyse board performance***

an activity that includes those sub-activities that relate directly to performance issues required by the customer. Results of analysis from these activities may be discussed during project development team meetings.

G.1.7**analyse board physical estimate**

an activity that includes those sub-activities that review the physical requirements of the electrical or electronic product. During project development team meetings, there is a review of the issues recorded during these activities.

G.1.8**analyse board reliability***

an activity that includes those sub-activities associated with determining the reliability of the electrical or electronic product based on customer environmental and other requirements.

G.1.9**analyse board specialty disciplines***

an activity that includes those sub-activities that address the overall product performance, environmental, functional, and physical characteristics. Results of these analyses are discussed during project development team meetings.

G.1.10**analyse component placement***

an activity that includes sub-activities that review placement of parts within or on a major structure of the electrical or electronic product. These activities review the thermal, producibility, vibratory performance and other characteristics of the product. Issues recorded are presented during meetings of the product development team.

G.1.11**analyse design and review**

an activity that includes sub-activities specifically concerned with the physical design of the electrical or electronic product. The significant physical parameters are reviewed and discussed during meetings of the project development team.

G.1.12**analyse detailed architecture***

an activity that includes sub-activities that review every aspect of the detailed architecture. Issues uncovered are presented during meetings of the project development team for resolution.

G.1.13**analyse electronic design***

an activity that includes those sub-activities used to be sure that the design meets customer requirements through various types of analysis. Some of these analyses include performance, manufacturability, and reliability.

G.1.14

analysed electronic design issues*

a collection of issues and solutions recorded during the analysis of the electronic design activity.

G.1.15

analyse other specialty disciplines*

an activity that covers analyses performed by various specialty disciplines associated with product performance, environmental, and other areas.

G.1.16

analyse producibility*

an activity that includes those sub-activities which research the relative ease with which this electrical or electronic product is produced. Issues uncovered during this analysis are presented to the project development team during its meetings.

G.1.17

analyse thermal properties*

an activity that performs thermal analyses on the electrical or electronic product and compares the results with customer requirements and specifications.

G.1.18

analyse vibration performance*

an activity that probes the performance of the electrical or electronic product under varying types of vibration. The results are compared to customer requirements and specifications. Issues discovered during these analyses are presented to the project development team during its meetings.

G.1.19

application tools*

software and hardware of all kinds which are common to the application environment and include software applications, spreadsheets, decision support tools, statistical analysis tools, project planning tools, and manual and automated design tools .

G.1.20

approved system design

the system or product architecture defined to support a customer or market need.

G.1.21

architecture concept issues*

a collection of issues and resolutions recorded during the definition of the detailed architecture concepts.

G.1.22

architecture function issues*

a collection of issues and resolutions defined during the validation of the detailed architecture functions of the product.

G.1.23**architecture manufacturability issues***

a collection of issues and resolutions recorded during analysis of board manufacturability.

G.1.24**architecture performance issues***

a collection of issues and resolutions recorded during analysis of board performance.

G.1.25**architecture physical estimate issues***

a collection of issues and resolutions recorded during analysis used to arrive at the physical estimates for the chosen architecture.

G.1.26**archival data**

the collection of all information which is maintained under strict configuration control and represents data required to build the subject PCA or PCB.

G.1.27**board definition**

a detailed enumeration of the board features including board outline.

G.1.28**board function issues***

a collection of issues and resolutions identified during analysis of board functions.

G.1.29**board performance estimate***

results which predict the degree to which the design in question will meet performance requirements.

G.1.30**board performance issues***

a collection of issues and resolutions identified during the performance estimating activity.

G.1.31**board physical estimate**

an estimate of the required physical needs of the product including the surface area required to accommodate all mounted components.

G.1.32**board physical issues***

a collection of issues and resolutions recorded during the physical estimating process for the board.

G.1.33

board requirements definition

a complete description of performance parameters for the bare board including both electrical and mechanical concerns.

G.1.34

build prototype

an activity that includes sub-activities focused on physical construction of the electrical or electronic product. Production issues may surface during these activities.

G.1.35

CAD data reference

the physical data file of the design.

G.1.36

compare material list with library

an activity that determines the viability of materials or physical resources needed to construct the product.

G.1.37

complete detailed architecture

an activity that includes sub-activities that determine if the architecture identified will satisfy the customer requirements or specifications. Included are validation, performance and manufacturability analyses.

G.1.38

complete design release

an activity that includes sub-activities that collect all pertinent project data associated with a specific electrical or electronic product and log that data into a formal configuration control system.

G.1.39

component placement data

data of one of the component placement priorities based on a functional placement approach which places important components integrated circuits or discrete in preferred locations according to board and product definitions.

G.1.40

component placement issues*

a collection of issues and resolutions recorded during component placement analysis.

G.1.41

control data*

information which governs enforcement of design rules required for routing of connections between components.

G.1.42**create electrical or electronic product**

within this activity are all of the sub-activities that accomplish product identification, design, fabrication, assembly, test and support of an electrical or electronic product. An activity that also creates new technologies that often accompany product creation and delivery.

G.1.43**create electronic design**

an activity that includes sub-activities that determine the functional interconnection of various technologies identified during the architecture design phase of the electrical or electronic product.

G.1.44**create functional test and simulation waves**

an activity that includes sub-activities that determine and create required test data.

G.1.45**create test and integrated diagnostic design***

an activity that includes sub-activities that generate and check testing strategies for the electrical or electronic product.

G.1.46**critical component issues***

a collection of issues and resolutions recorded during placement of critical components.

G.1.47**critical component placement data**

information identifying the location of specially placed parts.

G.1.48**customer operational requirements***

specifications addressing the performance of the required product, as defined by those desiring the fabrication and use of the product.

G.1.49**customer possible maintenance scenarios***

customer identified types of maintenance expected to be performed on the PCA or PCB.

G.1.50**customer requirements**

the highest level aggregation of requirements for the new product. May be that of a single customer or a market.

G.1.51**define board requirements**

an activity that includes sub-activities that address electrical or electro-mechanical requirements of the major parts of the electrical or electronic product.

G.1.52

define detailed architecture concept*

an activity that includes sub-activities that explicitly define the concepts used in meeting the requirements of the customer specifications.

G.1.53

define detailed architecture structure

an activity that identifies the hierarchy of the functional units used in satisfying the customer requirements of this electrical or electronic product.

G.1.54

define electronic design

an activity that includes sub-activities that determine in detail the interconnection of various functional blocks and their relationship to each other. Parts libraries are identified and test scenarios and test data are generated.

G.1.55

define electronic design issues*

a collection of issues and resolutions defined during the definition of the electronic design of the product.

G.1.56

design automation tools*

systems used to automate, accelerate, or accomplish the design process.

G.1.57

design mechanical PCA

an activity that includes sub-activities that identify, determine and verify the mechanical needs of the electrical or electronic product.

G.1.58

design rules data

information which governs the interconnection of components which have special requirements.

G.1.59

detail product definition development issues*

a collection of issues and resolutions recorded during detail product definition development activities.

G.1.60

detailed architecture

a complete description of the functional structure of the product to be designed and built.

G.1.61

detailed architecture concept*

a description of preliminary detailed architecture, physical, and performance estimates for the design.

G.1.62**detailed architecture issues log***

a collection of issues and resolutions recorded during the activity of developing the initial production plan.

G.1.63**detailed architecture issues***

a collection of issues and resolutions recorded during the activity associated with completing the detailed architecture and development of the initial production plan.

G.1.64**detailed architecture physical estimate***

an assessment of the physical resources required to accomplish the desired architecture.

G.1.65**detailed architecture structure**

a specific logical and functional definition of the electronic product.

G.1.66**detailed parts library issues***

a collection of issues and resolutions recorded during development of a detailed parts library.

G.1.67**determine general design rules***

an activity that includes sub-activities necessary to ascertain and acquire the necessary design rules to be used in routing the PCB.

G.1.68**determine other routing control data**

an activity that includes sub-activities necessary to ascertain and acquire the necessary control data of a more specific nature to the PCB undergoing routing.

G.1.69**determine special design rules**

an activity that includes sub-activities necessary to ascertain and acquire the specific routing requirements for any critical or special geometry or technically critical control information pertaining to the PCB undergoing routing.

G.1.70**determine special routing features**

an activity that includes sub-activities necessary to accommodate special features of the PCB such as keep out areas, cut-outs or special symbols.

G.1.71

develop drawing set

an activity that includes sub-activities necessary to produce engineering drawings that describe the electrical or electronic product in enough detail for production within the enterprise or at other facilities.

G.1.72

develop prototype test*

an activity that focuses on producing a test plan that, when executed, will fully exercise the prototype electrical or electronic products. This test plan forms the basis for future production testing activities.

G.1.73

develop detail product definition

an activity that includes all sub-activities associated with the physical design of the electrical or electronic product.

G.1.74

develop detailed architecture*

these activities include sub-activities which refine the electrical or electronic product architecture. These activities include definition of the structure, concept, and production plans. Analysis of the resulting architecture is performed during an activity of that segment.

G.1.75

develop detailed parts library*

an activity that includes sub-activities that develop, update or otherwise refine parts libraries associated with the electrical or electronic products.

G.1.76

develop initial production plan*

an activity that includes sub-activities that define how the electrical or electronic products will be produced. A draft production plan is created for review and discussion by the full product development team.

G.1.77

develop preliminary product definition

an activity that includes the sub-activities that define the architecture and electronic design of the product. An activity that further performs requirements analyses that are compared to customer functional specifications.

G.1.78

develop product

an activity that includes all the engineering activities that perform electrical or electronic (functional) design, physical design, and those analyses that simulate the behaviour of the unit to verify aspects of performance of the design in its intended environment. All sub-activities required to transform the conceptual design into a manufacturable product are within this activity.

G.1.79**develop structural tests***

an activity that includes those sub-activities associated with testing functional or logical structure of the product.

G.1.80**discrete issues***

a collection of issues and resolutions recorded during placement of discrete components to be used during the routing and layout of the PCB.

G.1.81**do conceptual design**

an activity that includes all of the sub-activities associated with transforming customer's requirements or request from ideas to a detailed conceptual design.

G.1.82**documentation issues***

a collection of issues and resolutions recorded during compilation of complete documentation of the design project.

G.1.83**draft detailed parts library***

an early version of the parts library for detailed design.

G.1.84**draft initial production plan***

a preliminary outline of the initial production plan.

G.1.85**draft preliminary product definition***

a pre-release version of the preliminary product definition. See preliminary product definition for additional information.

G.1.86**draft preliminary test and integrated diagnostic plan***

a draft of the preliminary test and integrated diagnostic plan. See preliminary test and integrated diagnostic plan for detailed definition.

G.1.87**drawing set issues***

a collection of issues and resolutions recorded during compilation of drawings associated with the development of the product.

G.1.88

electrical or electronic product*

the net result of satisfying requirements for a specific device, designed and manufactured according to standards, policies and procedures of one or more enterprises.

G.1.89

electronic design

the functional circuit design of the product with no physical layout information attached.

G.1.90

electronic design issues*

a collection of issues and resolutions recorded during the creation of the electronic design.

G.1.91

engineering drawings*

the graphical representation of the developed design with textual annotations as required.

G.1.92

engineering release notice*

document that authorizes the collected package of documents, parts lists, data, etc., to be released to manufacturing under configuration control.

G.1.93

engineering released documentation set

final approved set of complete documentation that represents a complete product design.

G.1.94

enhanced component packing data

the optimized placement information for components based on component shape and size.

G.1.95

enterprise facilities and equipment

the aggregation of capital equipment, buildings, and other support items necessary to carry out the goals of the enterprise.

G.1.96

enterprise personnel*

the total of the personnel employed at the facility producing the product.

G.1.97

enterprise policies, rules and standards*

all policies, procedures and practices acknowledged by the enterprise as well as designation of materials, parts, and such processes, which are used repetitively by the enterprise.

G.1.98**evaluate material physical characteristics**

an activity that includes sub-activities that determine the proper materials for construction of the electrical or electronic product.

G.1.99**evaluate functional mechanical characteristics**

an activity that includes sub-activities that review customer requirements and the demands they will place on the electrical or electronic product associated with mechanical or physical constraints.

G.1.100**examine mechanical requirement**

an activity that includes sub-activities that analyse the customer requirements from a physical or structural view point. Special mechanical requirements are identified and addressed.

G.1.101**execute board routing**

an activity that includes sub-activities that execute the physical connections necessary to realize the functionality of the electrical or electronic product.

G.1.102**fabrication data***

data necessary to fabricate a prototype of the electronic or electrical product.

G.1.103**finish detailed PCA definition**

an activity that includes sub-activities associated with completing the physical design of an electrical or electronic product not specifically addressed during the mechanical activities. Some of these activities include further update of the parts library, placement of parts on major sub-assemblies, analysis of placement, and refinement of component arrangements.

G.1.104**functional mechanical characteristics issues***

a collection of issues and resolutions recorded during evaluation of mechanical characteristics of the product.

G.1.105**functional test and simulation waves***

the simulation waveforms used to verify the functional performance of the circuit in a simulation.

G.1.106**generate material information***

an activity that includes sub-activities that provide information on available resources.

G.1.107

IC issues*

a collection of issues and resolutions recorded during the placement of integrated circuits on the PCB.

G.1.108

identify product*

an activity that includes all of the sub-activities required to identify new products. The activities are initiated from within the enterprise or as the result of an external customer request or requirement.

G.1.109

initial production plan*

the production plan intended to make a prototype PCA from an approved physical design.

G.1.110

locate components

an activity that defines the relative placement and location of all components used in the electrical or electronic product.

G.1.111

make prototype

an activity that includes sub-activities that allow for the construction and testing of prototype electrical or electronic products focused on meeting customer requirements and specifications.

G.1.112

make prototype issues*

a collection of issues and resolutions recorded during construction of prototype products.

G.1.113

manufacture product*

an activity that includes sub-activities that install components in or on the electrical or electronic product to complete the intended design.

G.1.114

manufacturing planners and production personnel*

those employees of the enterprise whose functions are to plan the processes to manufacture products and those employees who execute the processes.

G.1.115

material library issues*

a collection of issues and resolutions recorded during comparisons of the material lists and library resources.

G.1.116

material physical characteristics issues*

a collection of issues and resolutions recorded during review of the material physical characteristics.

G.1.117**mechanical board design**

generated mechanical data, physical concept, characteristics and constraints, chassis measurement, physical constraints, plus identified limits on temperature, humidity, vibration, and EMI needed to completely describe the mechanical configuration of the product.

G.1.118**mechanical configuration**

the mechanical aspect of the product that includes spatial requirement, chassis measurement, physical constraints for PCA boards (limits on width, length, height, tolerance, clearance), environmental and working requirements, limits on temperature, humidity, vibration, and EMI.

G.1.119**mechanical parts list**

a list of parts required for the PCA which are necessary in addition to those parts used to accomplish the electrical functionality of the design.

G.1.120**mechanical PCA issues***

a collection of issues and resolutions recorded during design of the mechanical aspects of the printed circuit assembly.

G.1.121**needs for design change***

descriptions of issues found in the manufacturing processes that appear to require changes in the PCA design for solution.

G.1.122**needs for facilities and equipment***

an identification and recognition of the necessity for additional resources in the form of technically innovative or new facilities or other equipment which will facilitate implementation of a new technology.

G.1.123**new parts request***

a request made by a designer or design team member to add or modify component information to an official library used by an enterprise.

G.1.124**new technology block concept***

the ideas and theory behind new areas of technology to be used in development of the product.

G.1.125**new technology block definition***

the description and requirements for a new technology to be used in development of the product.

G.1.126

optimize packing density

an activity that refines the relative placement and location of all components used in the electrical or electronic product in order to minimize overall volume of the product.

G.1.127

optimized routed board

the last iteration of board routing and analysis which represents a satisfactory solution to the issues of routing.

G.1.128

PCA definition issues*

a collection of issues and resolutions recorded during completion of PCA definition.

G.1.129

perform calculations*

an activity that involves the calculations that must provide additional information to higher sub-assemblies that may incorporate the electrical or electronic product being designed. Such calculations may include mass properties, center of gravity, radiation analysis, and others.

G.1.130

place component issues*

a collection of issues and resolutions recorded during the placement of components on the printed circuit board.

G.1.131

place components

an activity that includes sub-activities that determine the position of components in and on sub-assemblies, constituting the basis for the electrical or electronic product.

G.1.132

place critical components

an activity that defines the relative placement and location of critical components used in the electrical or electronic product.

G.1.133

place discrete components*

an activity that defines the relative placement and location of discrete components used in the electrical or electronic product.

G.1.134

place IC components

an activity that defines the relative placement and location of IC components used in the electrical or electronic product.

G.1.135**placement analysis data**

data obtained after analysing component placement to assure compatibility to preliminary product definition and physical concept.

G.1.136**placement data**

information about the placed components on a printed circuit board.

G.1.137**placement parameters**

factors which set the proper operating environment for placing components on a printed circuit board.

G.1.138**preliminary parts library**

the initial parts library that is available to support electrical design.

G.1.139**preliminary parts list**

a list of parts that the project or organization has designated as preferred parts during product identification.

G.1.140**preliminary product definition**

the net-list, schematic, board outline, connectors, and estimates of area, performance, and thermal characteristics which meet initial requirements of the product.

G.1.141**preliminary routed board data**

the information resulting from a first pass or gross routing of the PCB.

G.1.142**preliminary test and integrated diagnostics plan***

an outline and detail of planned tests. Details on integrated diagnostics are identified for review and use with this plan.

G.1.143**prepare for placement**

an activity that includes sub-activities that coordinate all necessary resources used in the positioning of components of the electrical or electronic product.

G.1.144**prepare for routing**

an activity that includes sub-activities necessary to gather all required resources, rules, requirements, and associated supporting entities needed in the routing process.

G.1.145

product concept*

documentation to define the new product which includes an architectural concept, maintenance concept, marketing concept, manufacturing concept, or support concept.

G.1.146

product data

all data that is used to describe aspects of a product.

G.1.147

product development issues*

a collection of issues and resolutions related to the product development phase.

G.1.148

product qualification issues*

a collection of issues and resolutions recorded during product qualification activities.

G.1.149

product support*

the process of providing product-specific and usage-specific technical information through documentation, on-line help, tutorial, training, and other methods to users of the product.

G.1.150

production release notice*

a formal communication indicating the design is released to the production activity.

G.1.151

production released product data set

the final collection of all data needed to produce the product that exist in a formal configuration control system.

G.1.152

project development team*

circuit designers, test engineers, physical designers, design engineers, manufacturing engineers, managers, and other enterprise resources assembled to accomplish a specific task.

G.1.153

qualify product*

an activity that includes sub-activities required by the enterprise to ensure full compliance with customer requirements have been met prior to release for production. This activity may also provide first article or proof of concept functions.

G.1.154

reassign references

an activity that updates the references designators for individual components placement data.

G.1.155**refine placement**

an activity that presents alternative and optimal placement of all components associated with the product. Any trade-offs that may affect product functionality are reported to the product development team.

G.1.156**refine routing**

an activity that is specifically focused on establishing the optimum physical connectivity of the electrical or electronic product. Concerns with space utilization are also addressed.

G.1.157**refinement issues***

a collection of issues and resolutions recorded during the placement refinement activity.

G.1.158**release engineering design**

an activity that includes sub-activities that prepare the necessary documentation, consolidation, and distribution of the completed design. These sub-activities formalize the design for release to manufacturing processes or external customers.

G.1.159**release engineering design issues log***

a collection of issues and resolutions gathered during activities associated with releasing the engineering design.

G.1.160**release preliminary definition**

an activity that includes sub-activities that prepare the necessary documentation, consolidation, and distribution of the preliminary design. These sub-activities formalize the design for preliminary release to manufacturing processes or external customers.

G.1.161**route board**

an activity that includes sub-activities associated with defining the physical interconnection of the electrical or electronic product.

G.1.162**routed board data**

data obtained from routing operation executed according to the board definition.

G.1.163**routing control data**

routing parameters that govern the way conducting paths are placed on a printed circuit assembly.

G.1.164

routing features data

information which governs the use and application of special routing features or special symbols.

G.1.165

routing issues*

a collection of issues and resolutions encountered during the routing operation, such as identification of un-routable traces, design rule violations, or high density routing areas.

G.1.166

routing layout data

the information governing the use and application of conducting paths on each layer of a printed circuit board.

G.1.167

spawn new technology

an activity that includes all of the sub-activities involved in creating new technology bases. New and advanced technologies are utilized that meet critical and demanding customer requirements.

G.1.168

specialty issues*

a collection of issues and resolutions recorded during analysis of the specialty disciplines.

G.1.169

support product*

an activity that includes sub-activities necessary to complete full life cycle support. These activities may include, repair, replacement, instructional, field modification, and disposal of the electrical or electronic product.

G.1.170

swap device or gates

an activity that refines layout by reorganizing the device or gate functionality to component assignment.

G.1.171

swapped device or gate data

the information which identifies final device or gate relationships with the required conductors.

G.1.172

technology library*

a collection of applicable detail PCA technology information which may be enterprise-specific, product-line-specific, product-specific, or technology-specific.

G.1.173

test prototype

an activity that includes sub-activities that determine the actual performance of a prototype. Proof of testing procedures are also verified during this testing phase.

G.1.174**test equipment requirements***

identified capabilities of test equipment necessary to insure the proper functioning of the subject PCA or PCB.

G.1.175**test and integrated diagnostics plan***

a plan intended to test a prototype PCB to assure that it can meet requirements of the original product specification.

G.1.176**unavailable mechanical parts list***

a list of mechanical parts that are required for the collection of the design which are not currently in any parts library.

G.1.177**unavailable parts list***

a parts list that identifies those parts that are required in a design, but are not available in the parts library.

G.1.178**untested prototype**

the result of physically assembling all required components called out in the design of the subject PCA or PCB before confirmation of working functionality.

G.1.179**update part library**

an activity that includes sub-activities that furnish the project part library with the latest information on resources needed in the physical realization of the architectural concept and required functionality.

G.1.180**updated mechanical parts library***

the mechanical parts library that contains newly added mechanical parts that are required by the product design.

G.1.181**updated parts list**

a complete list of parts and quantities necessary to complete the product design.

G.1.182**validate detailed architecture function***

an activity that includes sub-activities that verify that the architectural concept presented for the electrical or electronic product is viable, sound, and technically possible.

G.1.183

validated mechanical configuration

the mechanical configuration that meets product spatial, environmental, and operational requirements for the product.

G.1.184

waves issues*

a collection of issues and resolutions associated with creation of functional test or simulation data.

G.2 Application activity model figures

The application activity model diagrams are given in figures G.1 through G.21. The graphical form of the application activity model is presented in the IDEF0 activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

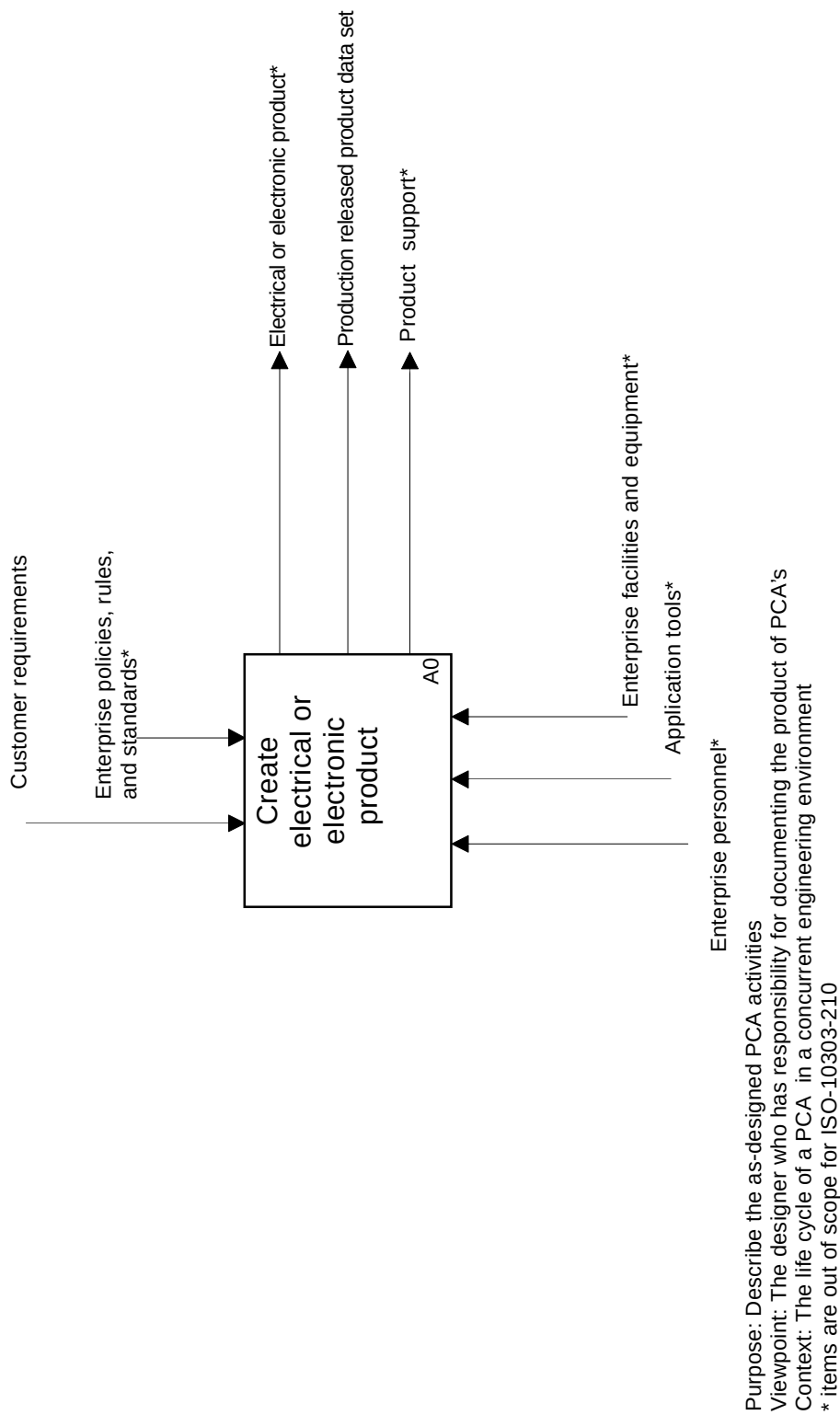


Figure G.1 – Electrical or electronic product life cycle (Context)

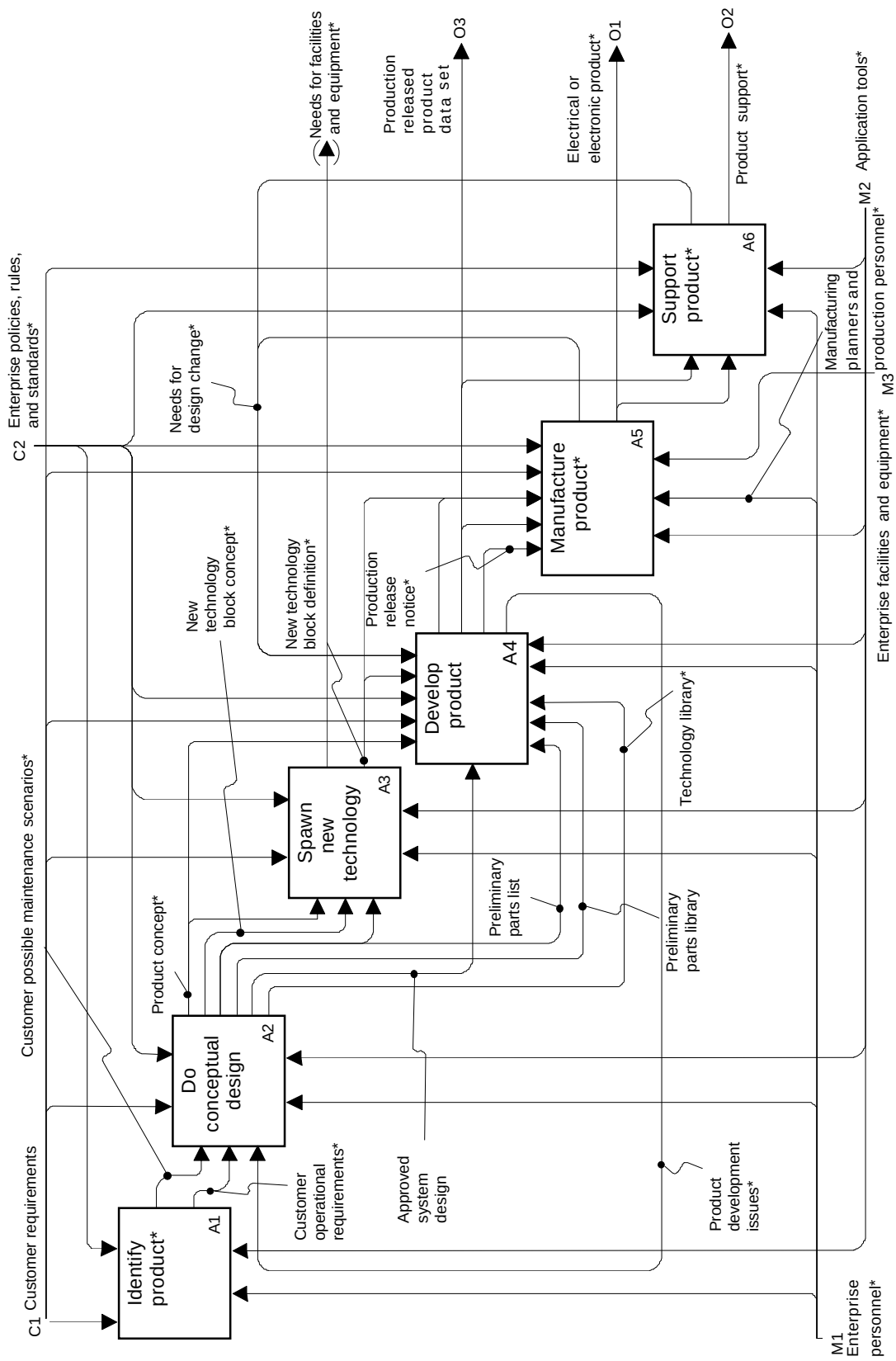


Figure G.2 – Create electrical-electronic product

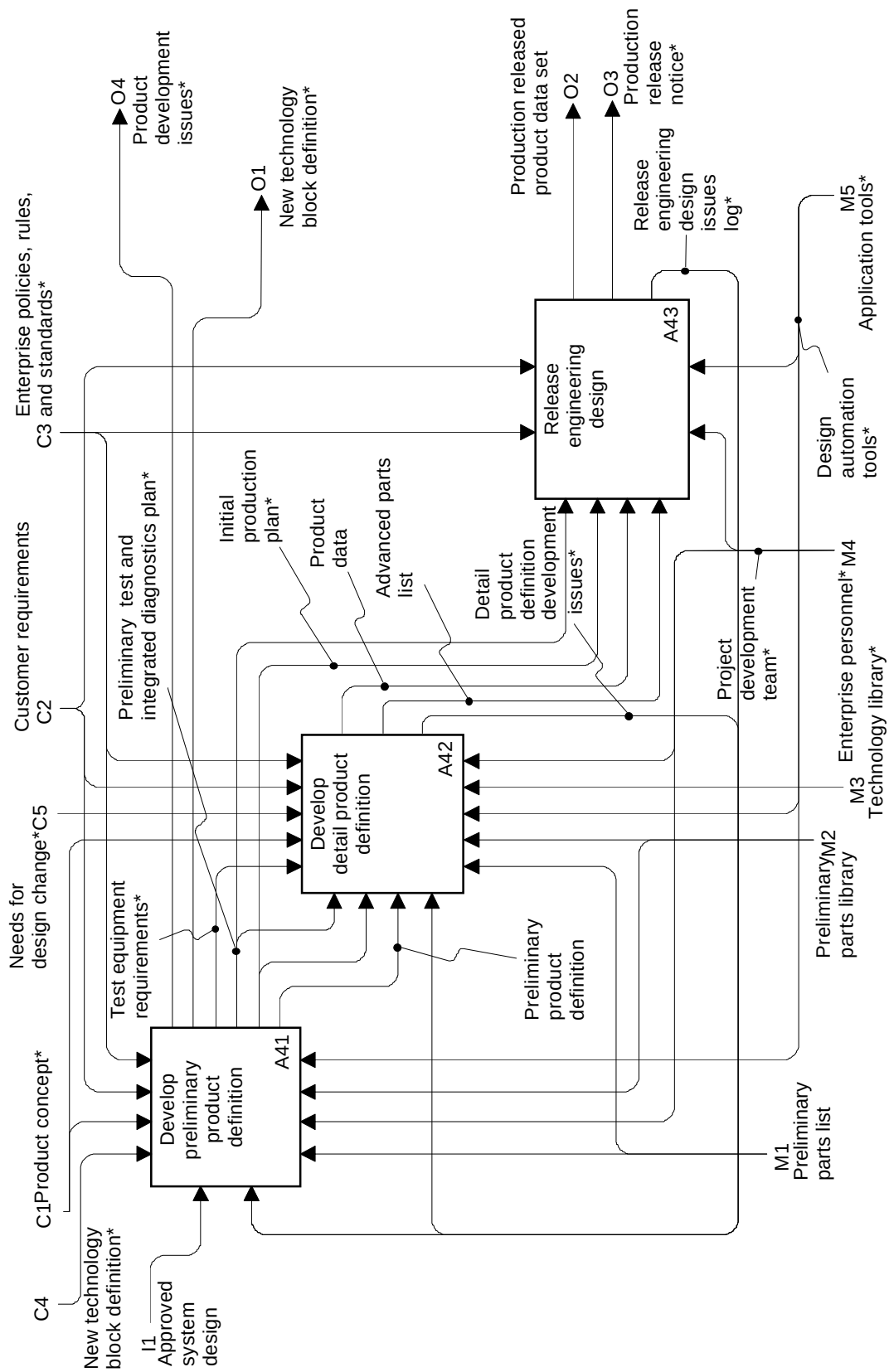


Figure G.3 – Develop product

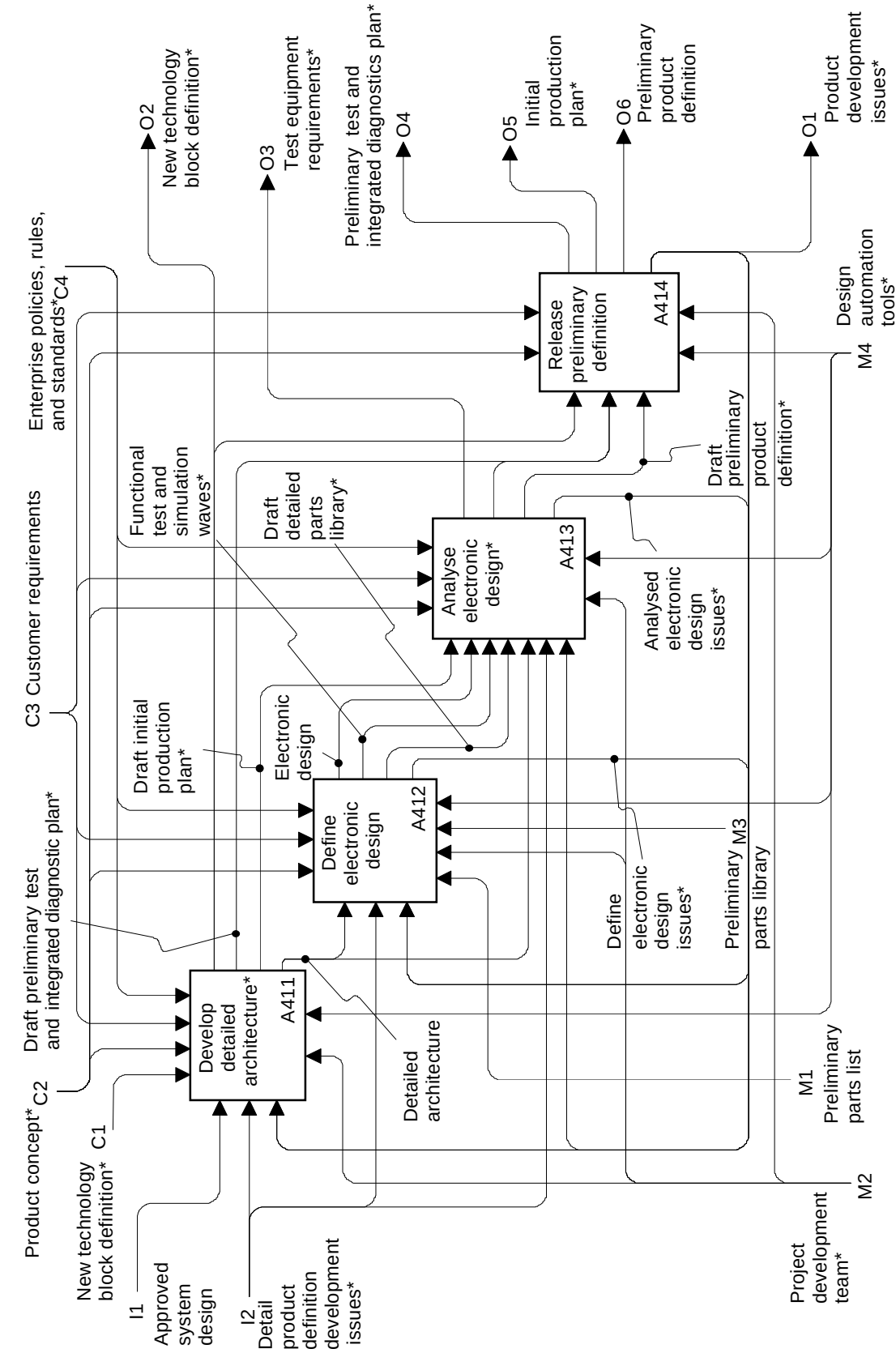


Figure G.4 – Develop preliminary product definition

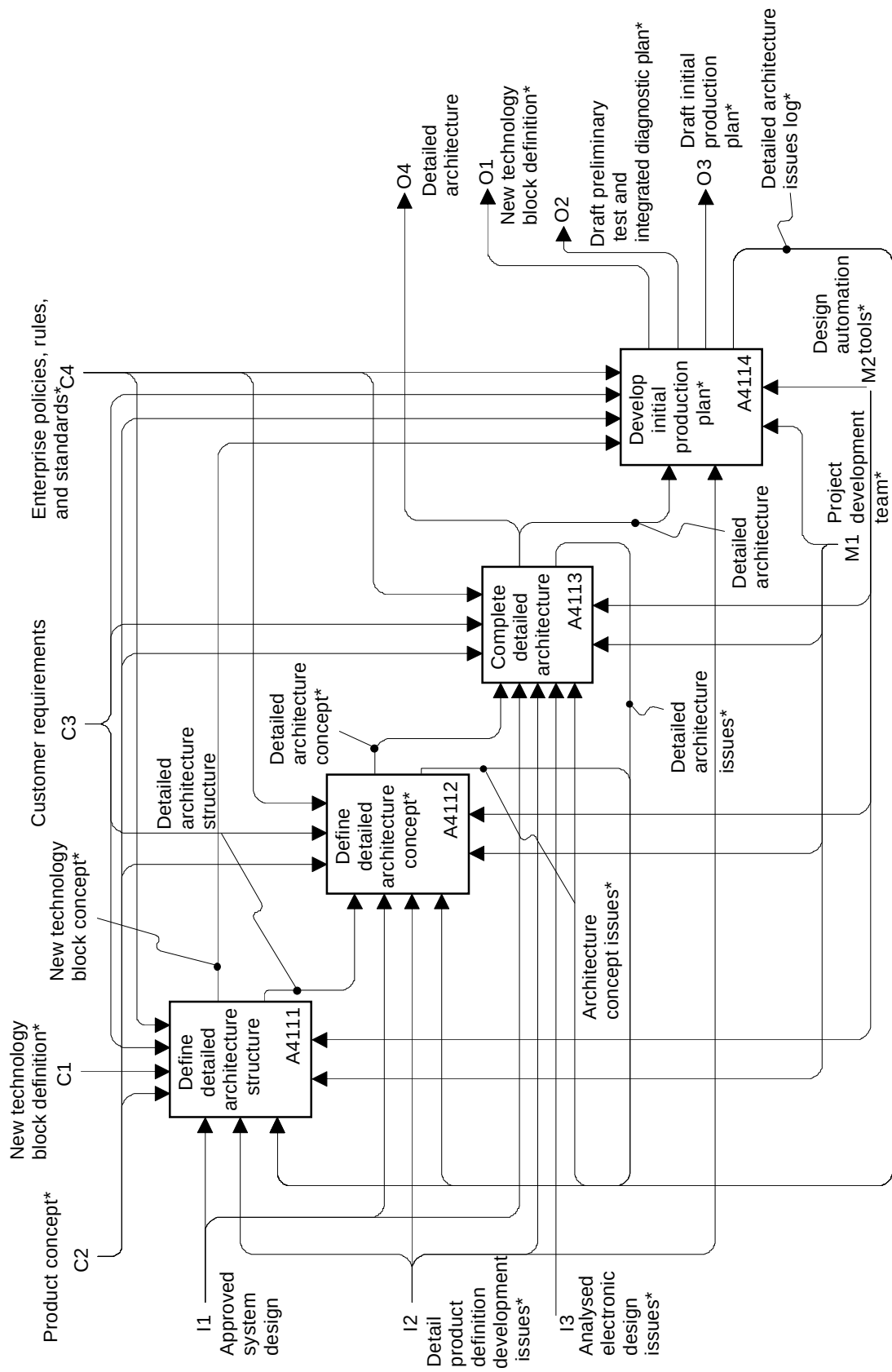


Figure G.5 – Develop detailed architecture

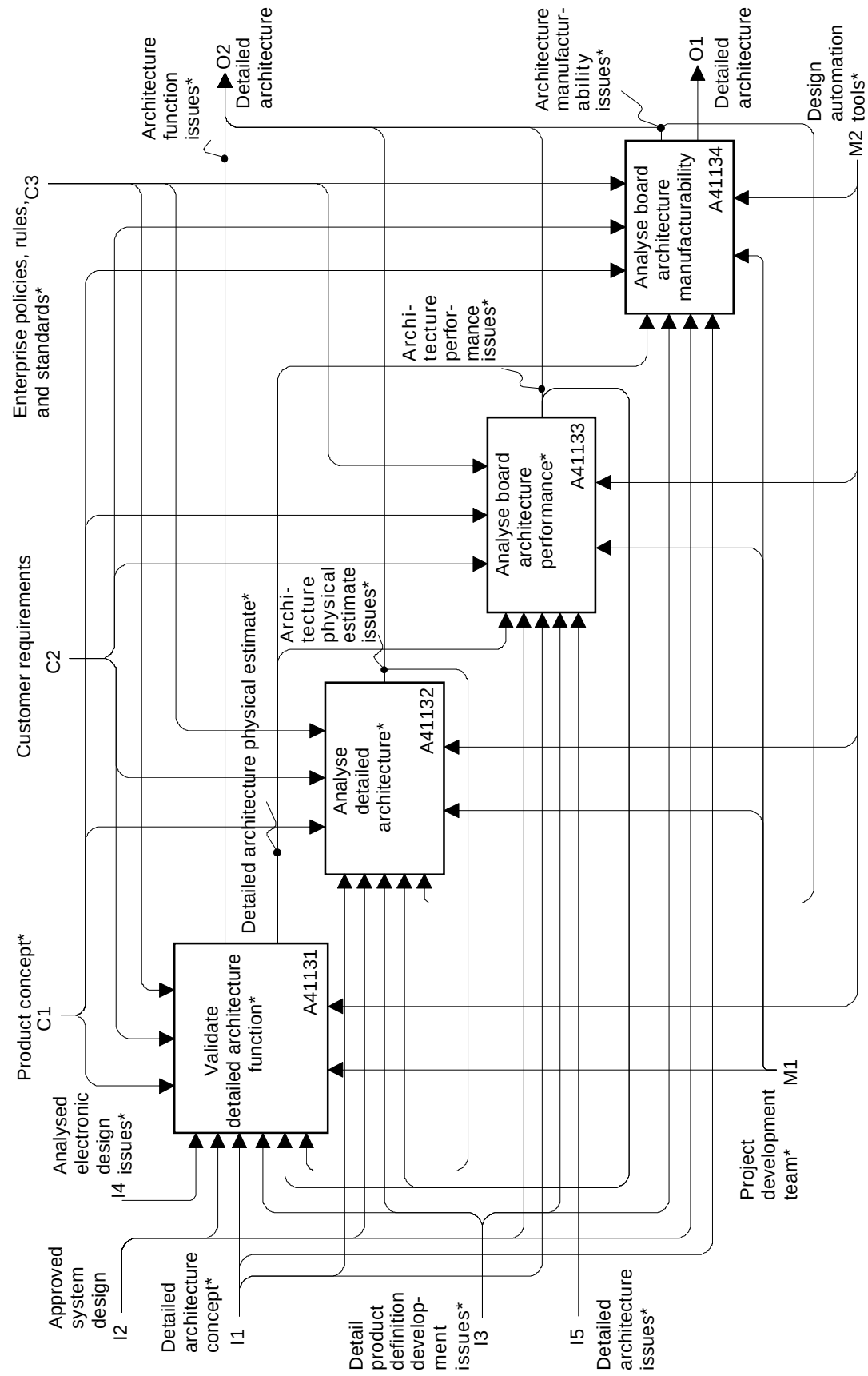


Figure G.6 – Complete detailed architecture

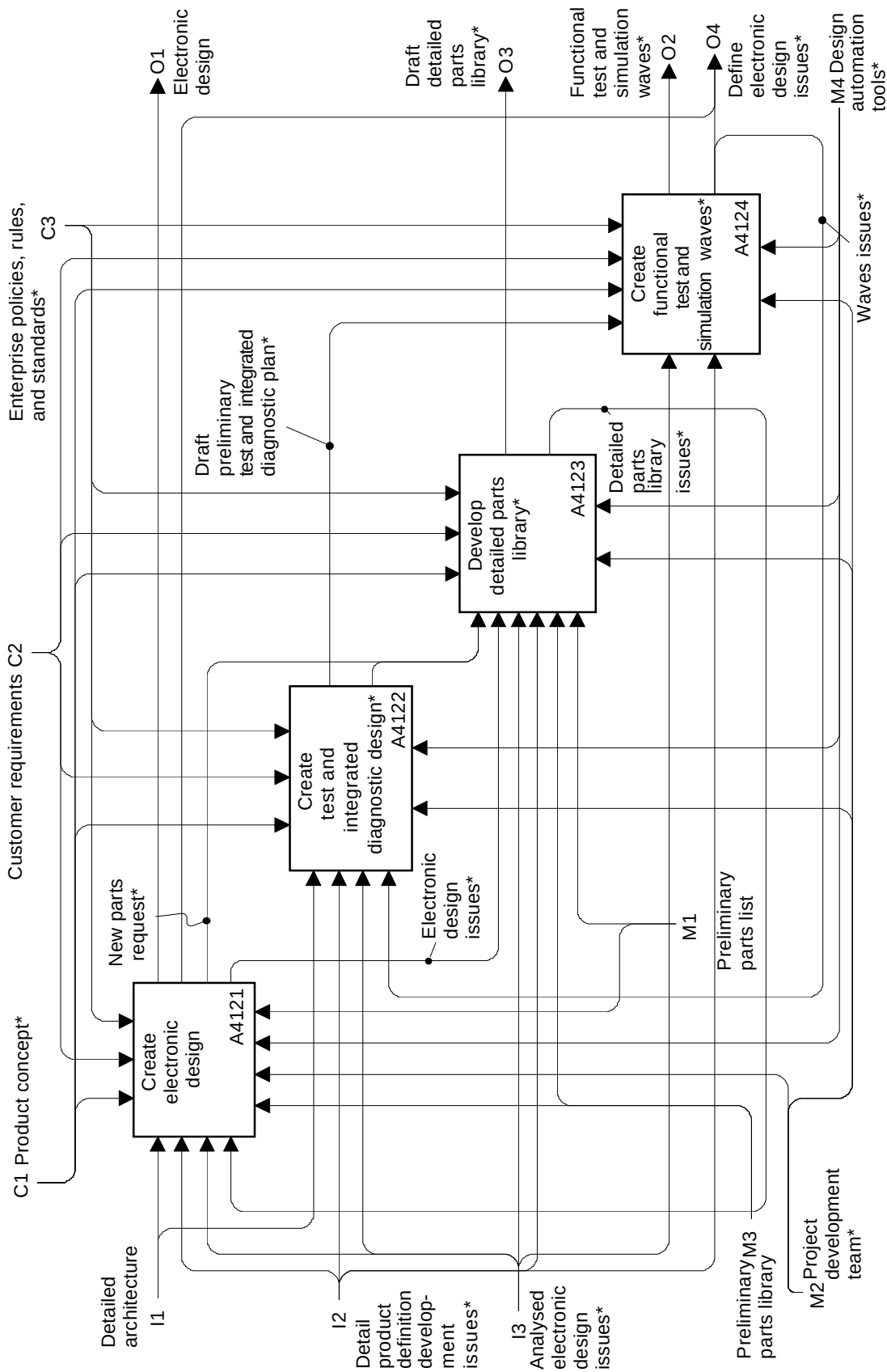


Figure G.7 – Define electronic design

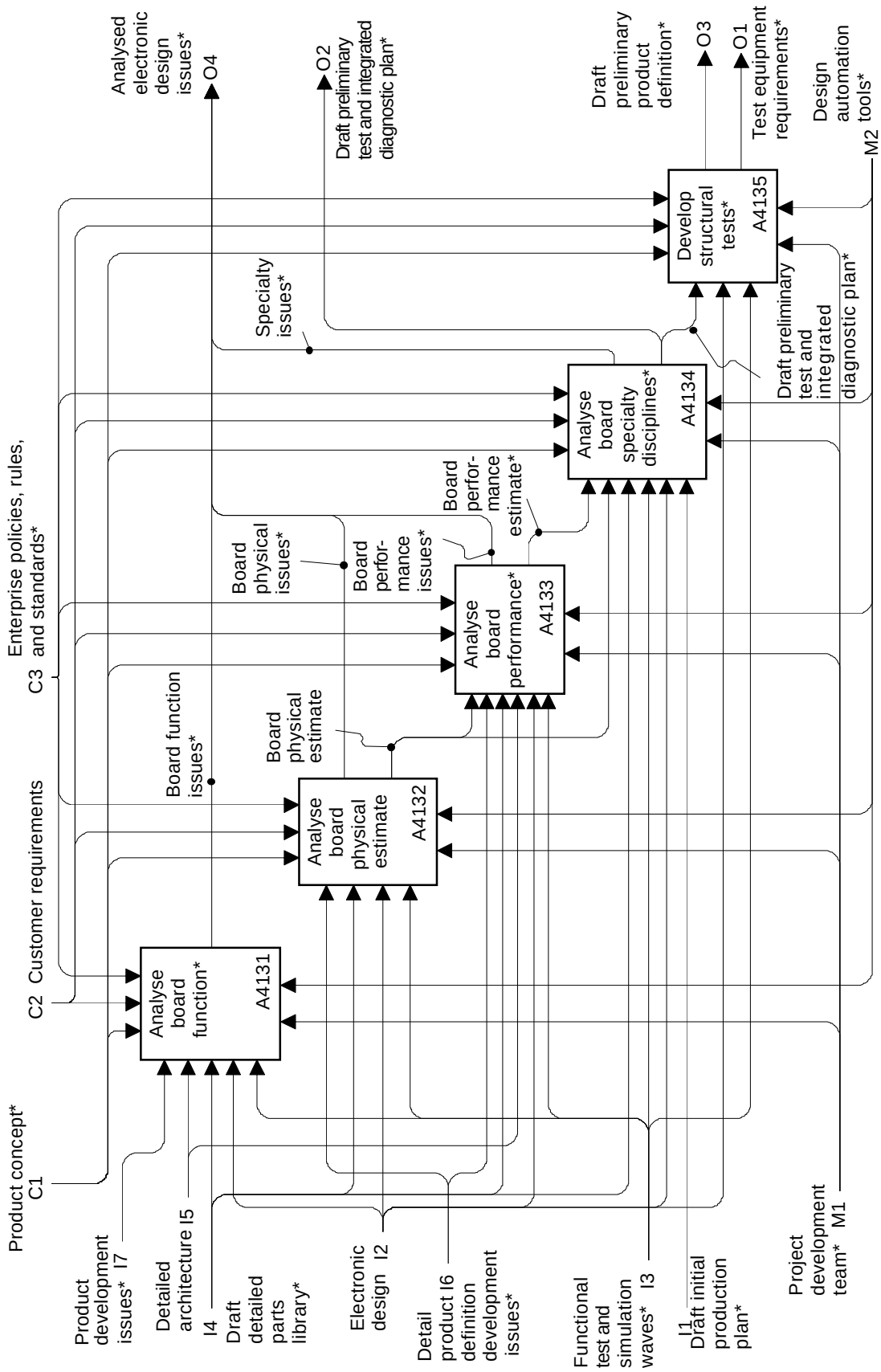


Figure G.8 – Analyse electronic design

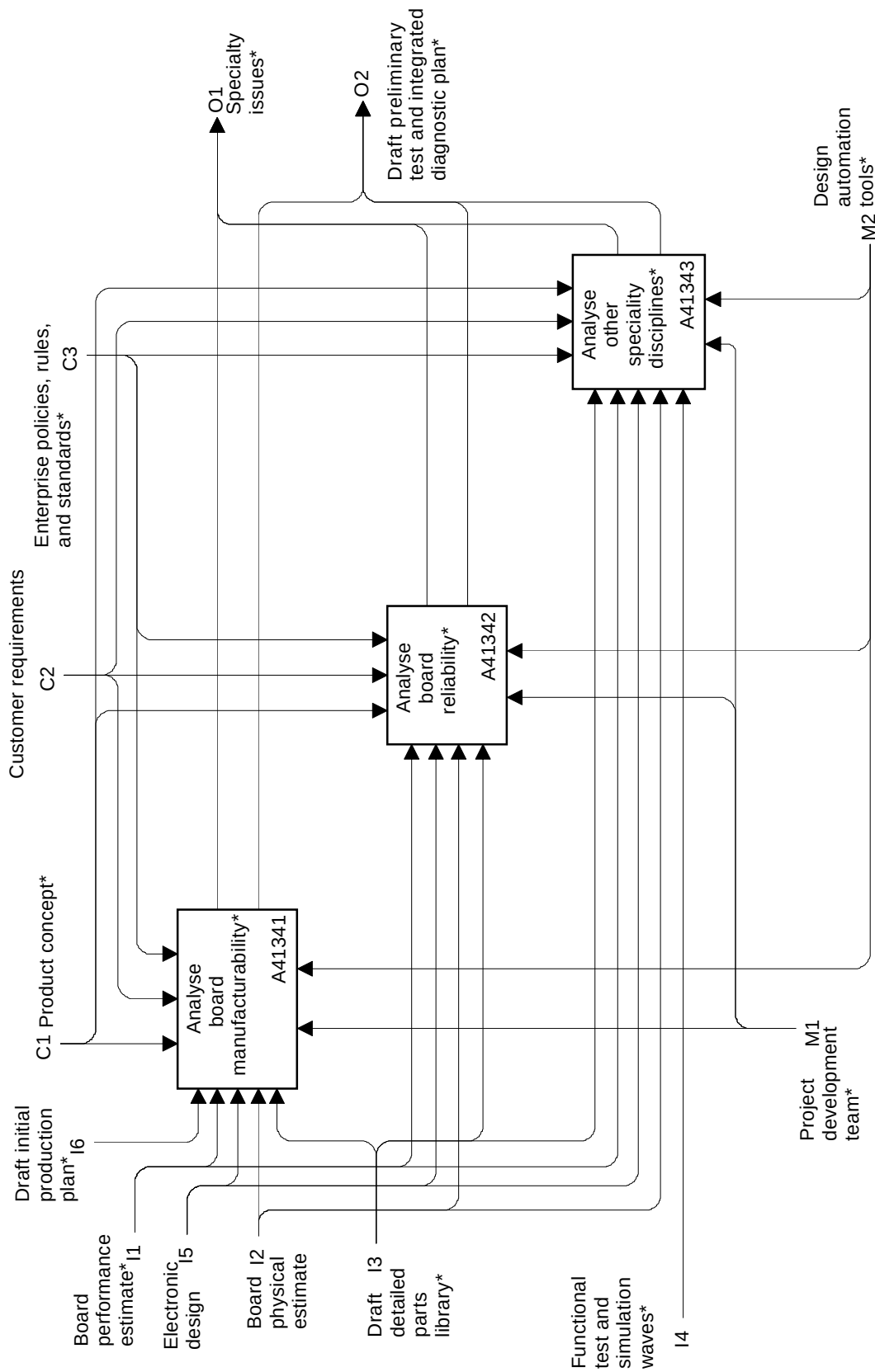


Figure G.9 – Analyse board specialty disciplines

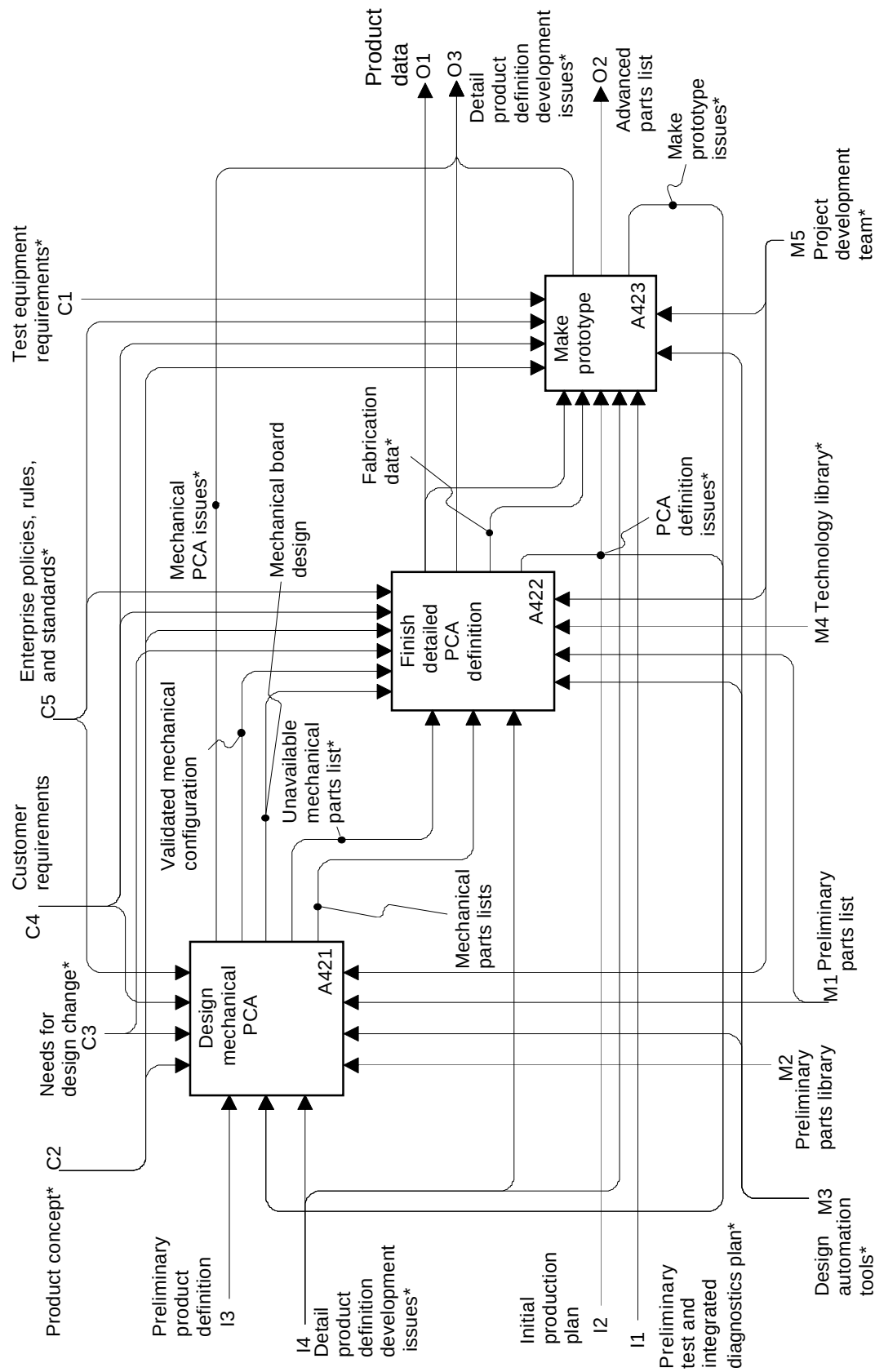


Figure G.10 – Develop detail product definition

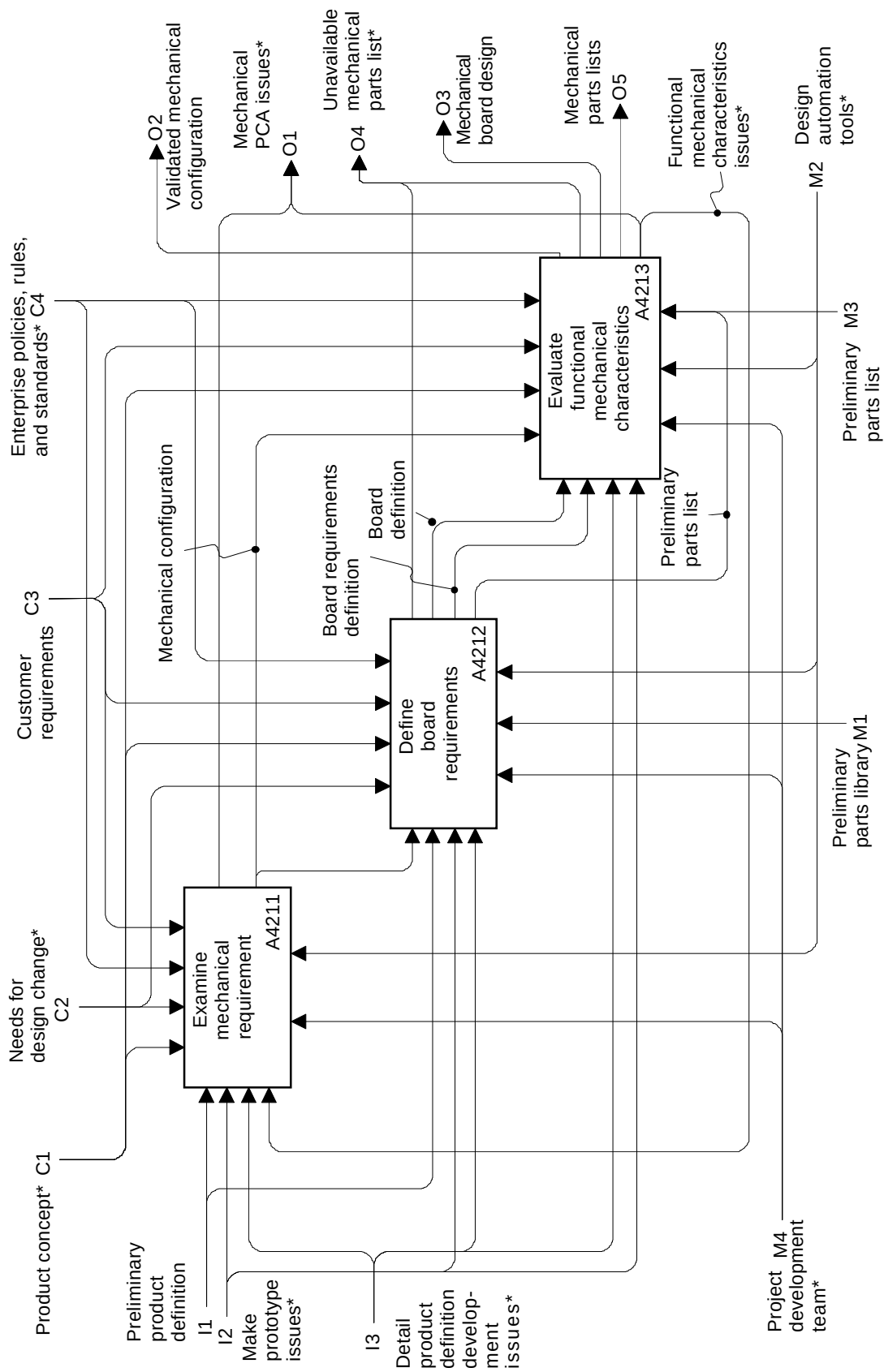


Figure G.11 – Design mechanical PCA

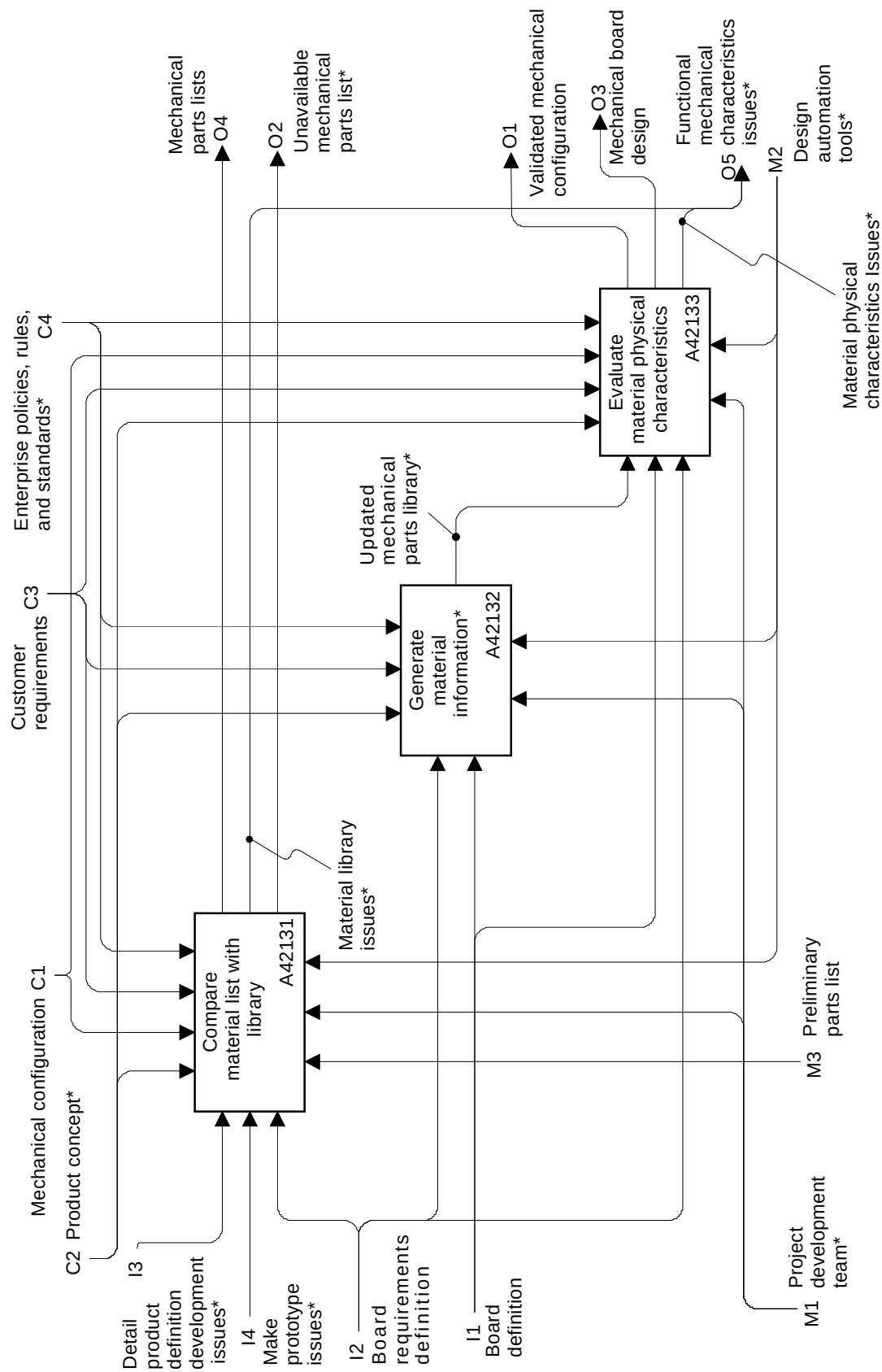


Figure G.12 – Evaluate functional mechanical characteristics

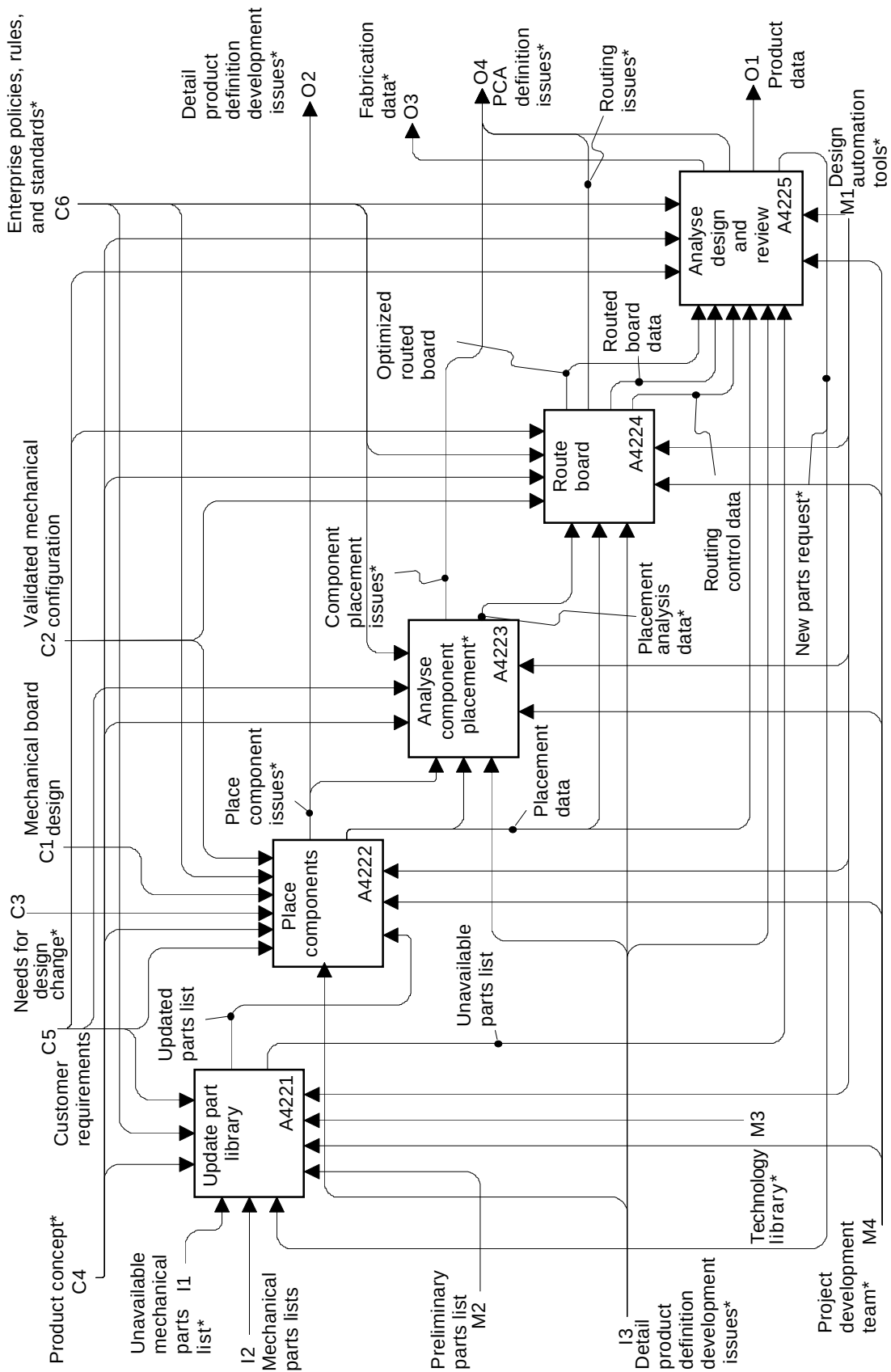


Figure G.13 – Finish detailed PCA definition

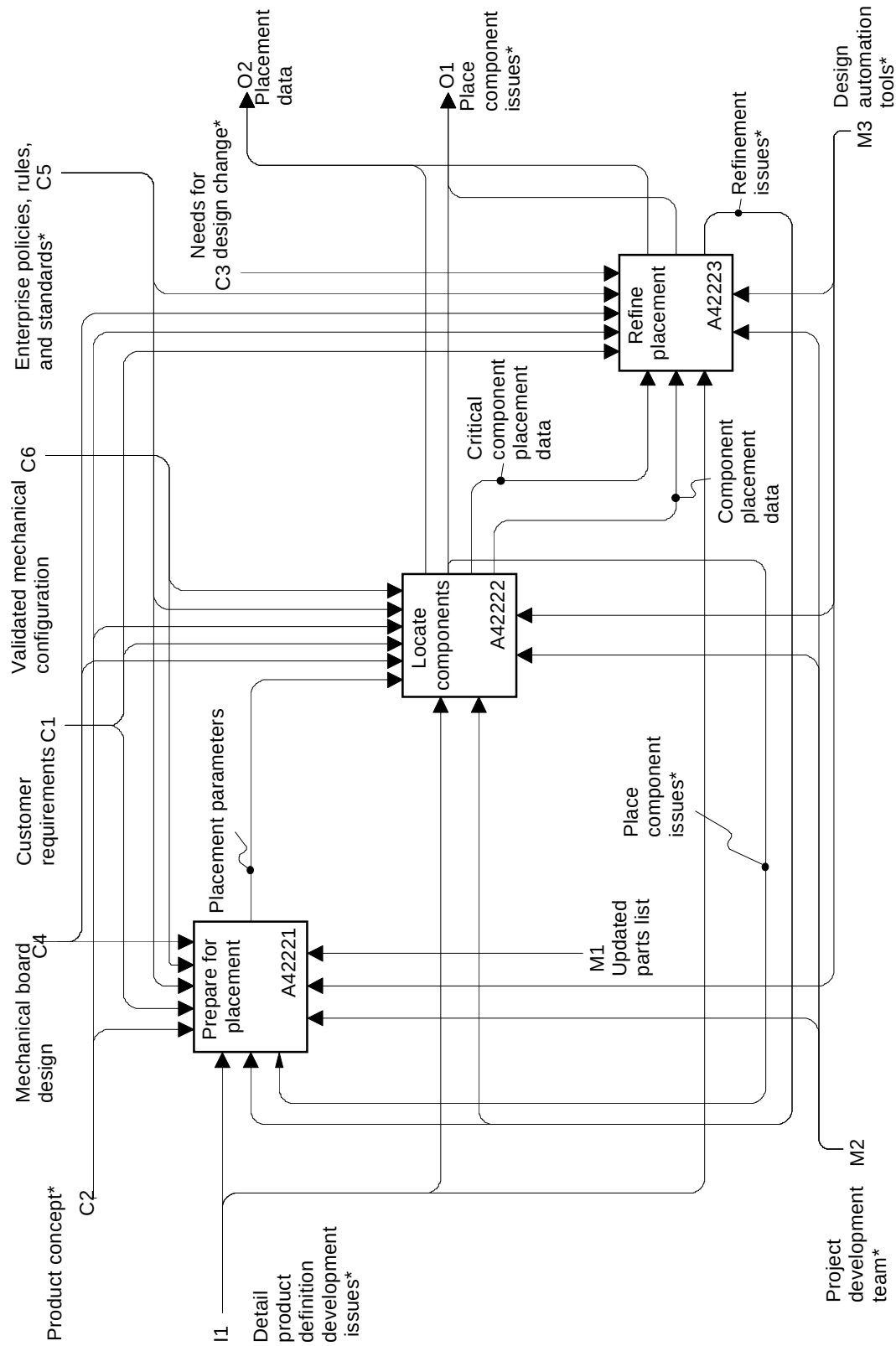


Figure G.14 – Place components

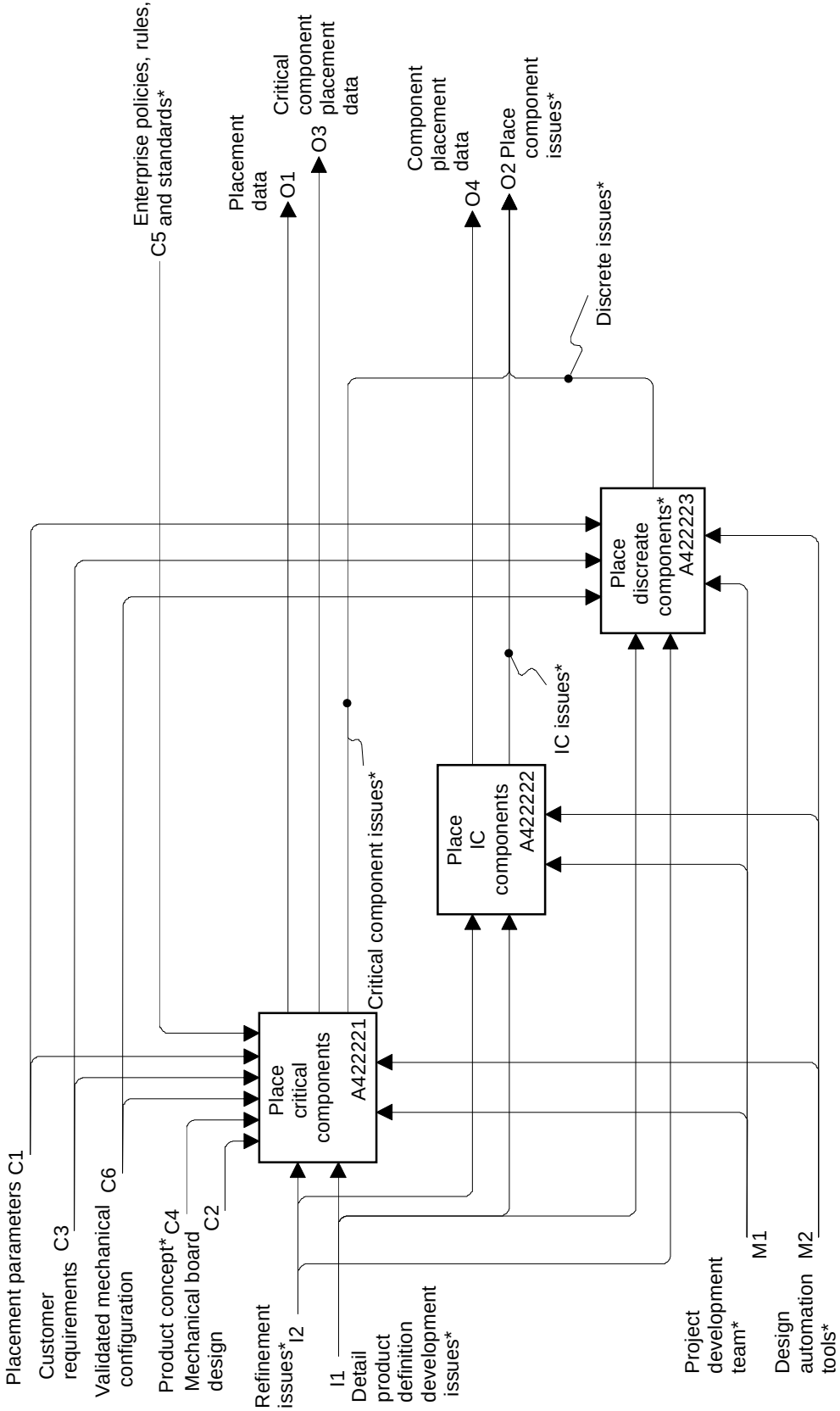


Figure G.15 – Locate components

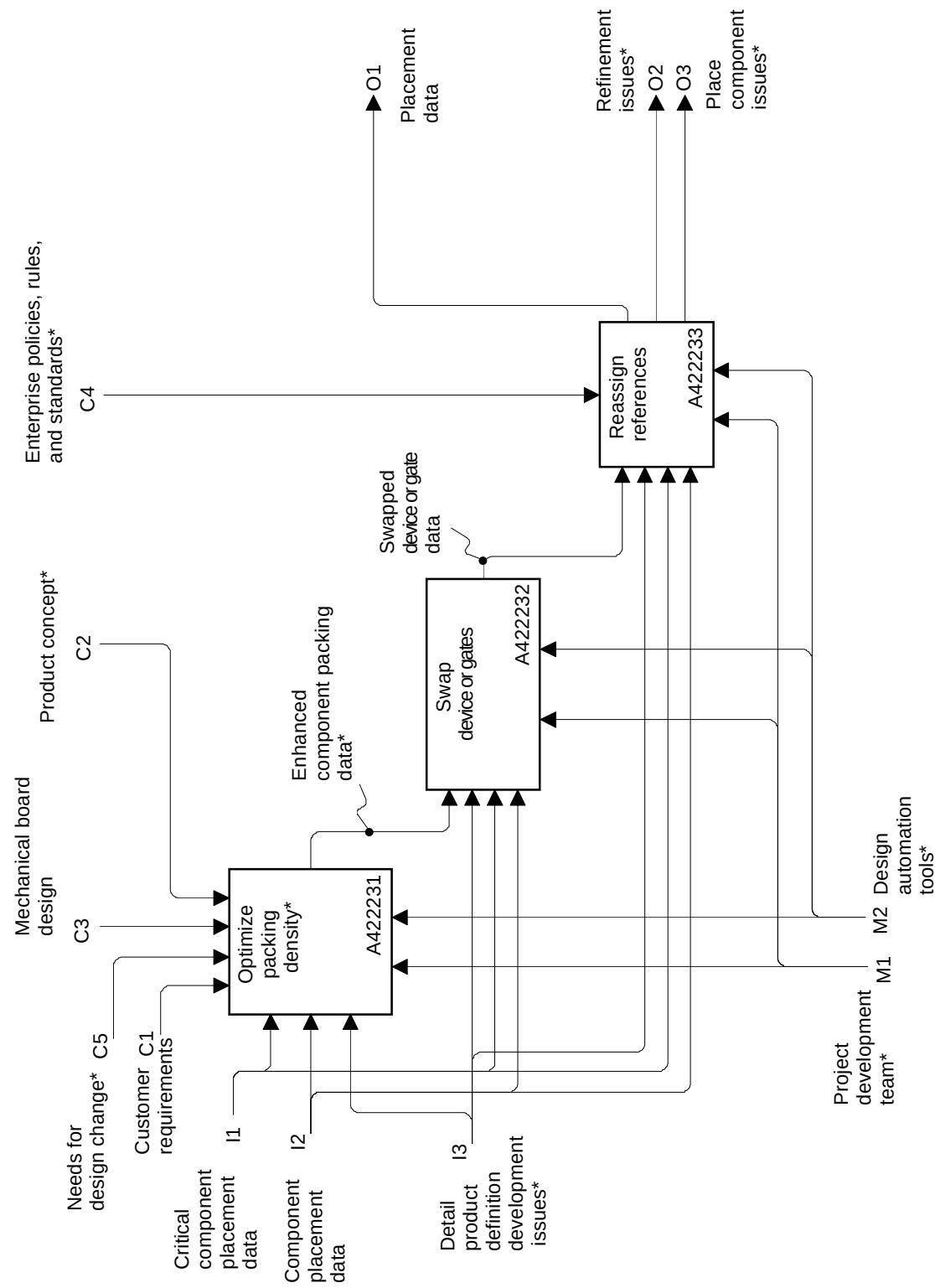


Figure G.16 – Refine placement

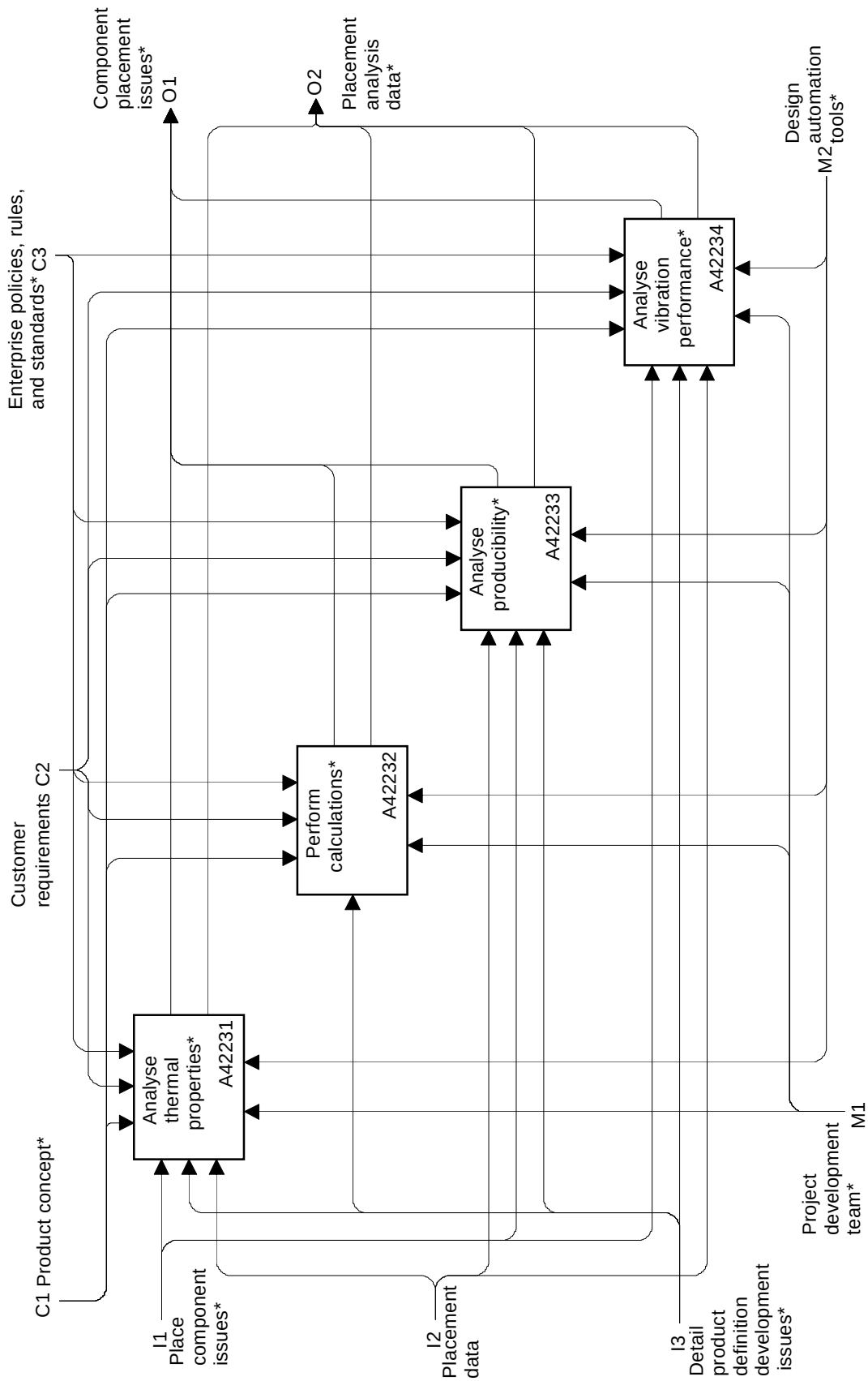


Figure G.17 – Analyse component placement

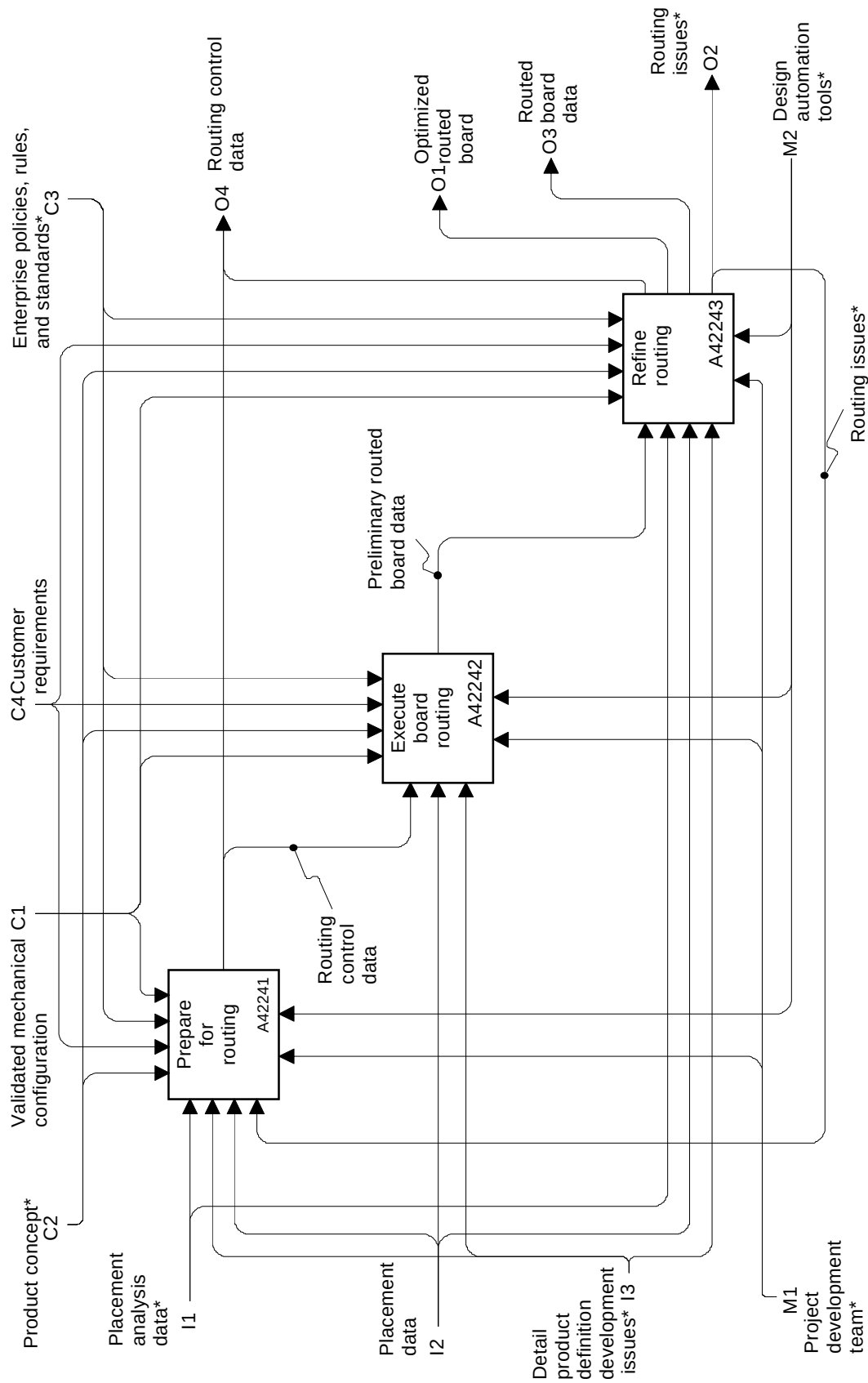


Figure G.18 – Route board

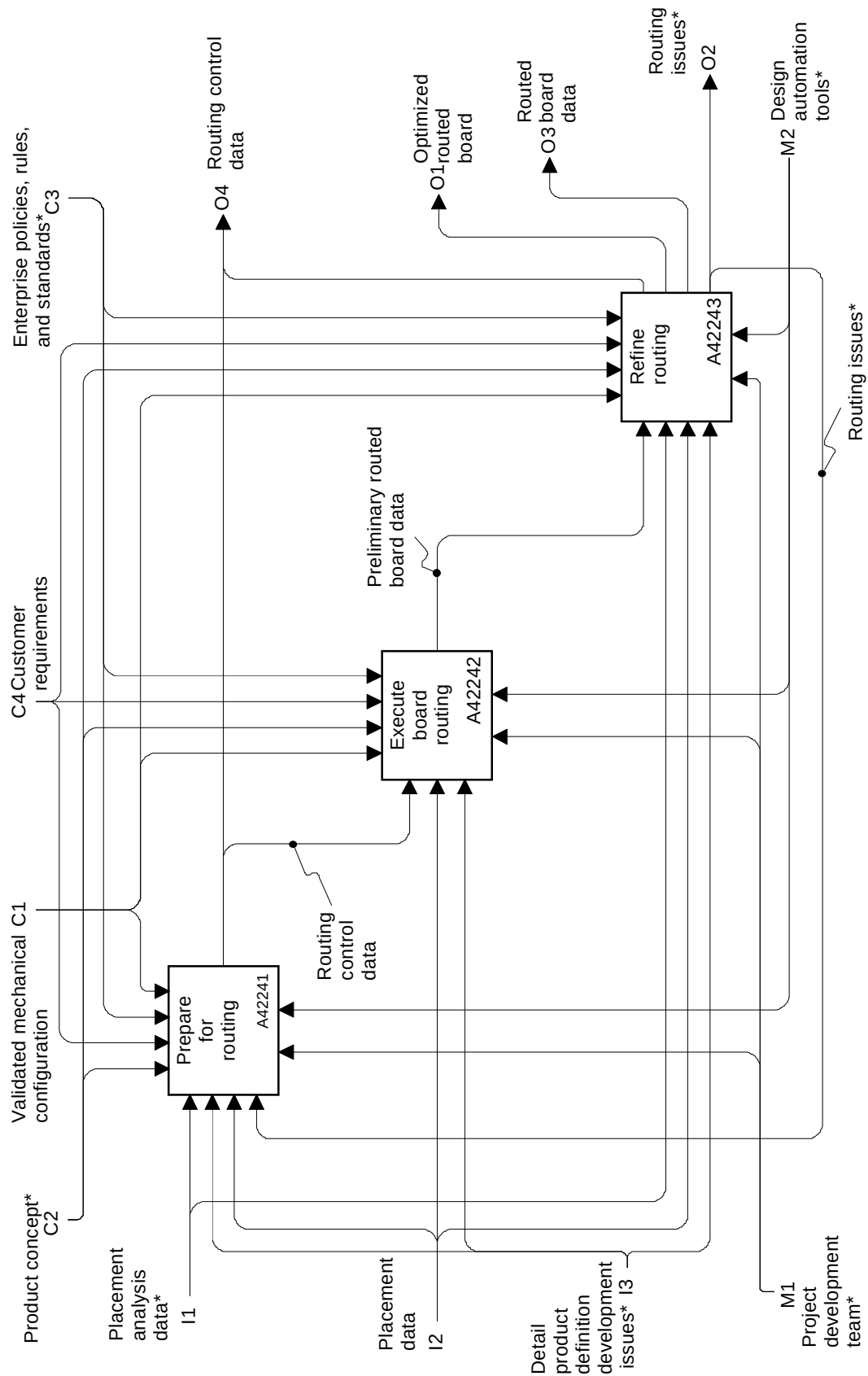


Figure G.19 – Prepare for routing

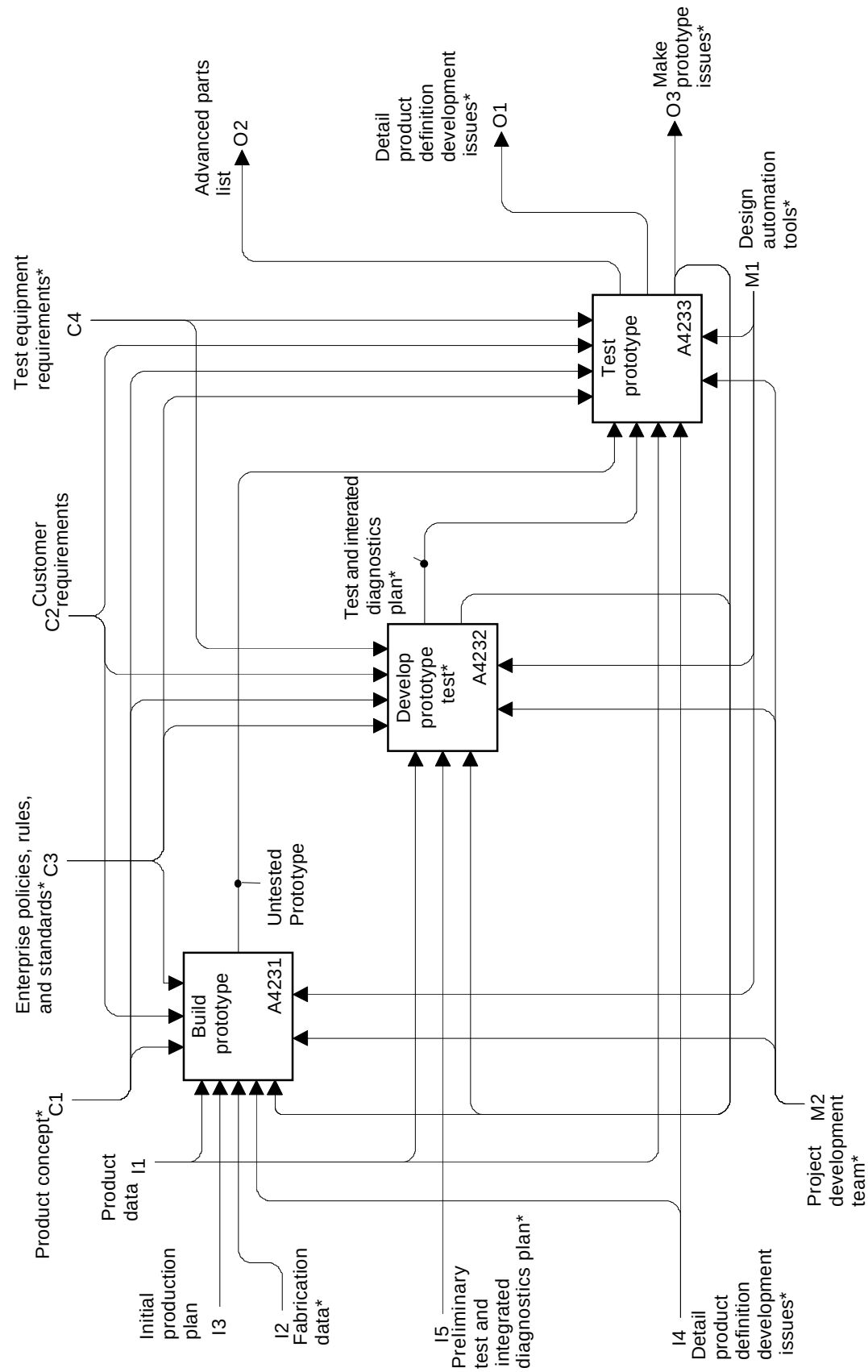


Figure G.20 – Make prototype

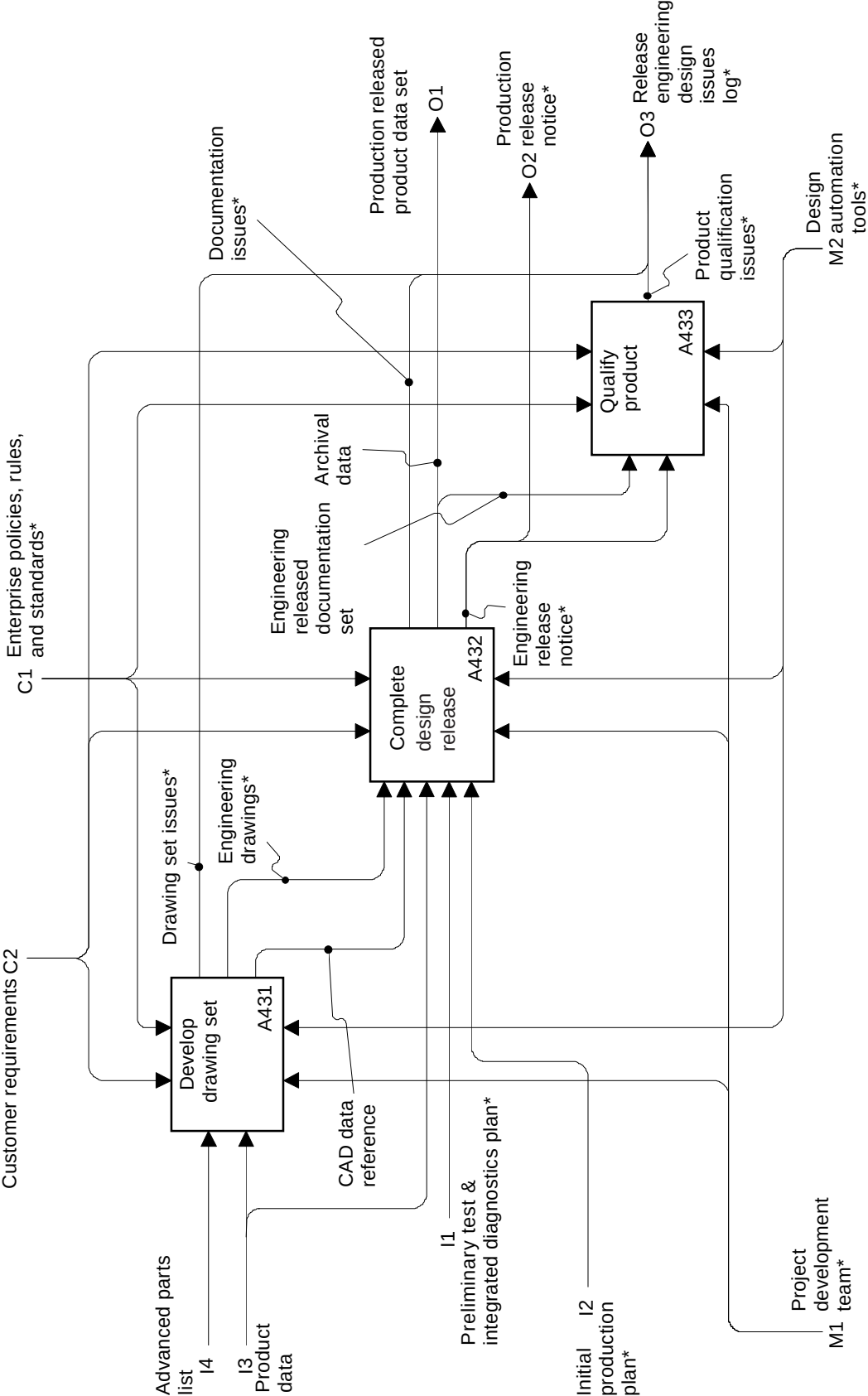


Figure G.21 – Release engineering design

Annex H (informative) Application reference model

This annex provides the application reference model for this Part of ISO 10303 and is given in figures H.1 through H.243. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in EXPRESS-G. The application reference model is independent from any implementation method.

H.1 EXPRESS

EXPRESS specification:

```
(*
This file was generated by the Java Express PrettyPrinter - alpha version
Written at the National Institute of Standards and Technology
by Stephane Lardet (stephane.lardet@nist.gov)
*)
```

```
SCHEMA AP210_ARM;
```

```
    TYPE ahead_or_behind = ENUMERATION OF
        (ahead,
         behind);
    END_TYPE; -- ahead_or_behind
```

```
    TYPE angle_relator = ENUMERATION OF
        (equal,
         large,
         small);
    END_TYPE; -- angle_relator
```

```
    TYPE assembled_feature_select = SELECT
        (interconnect_component_join_terminal,
         interconnect_component_interface_terminal,
         packaged_connector_component_interface_terminal,
         package_terminal_occurrence,
         bare_die_component_terminal,
```

```

    packaged_component_join_terminal,
    assembly_module_component_terminal,
    interconnect_module_component_surface_feature,
    interconnect_module_component_terminal,
    component_mounting_feature);
END_TYPE; -- assembled_feature_select

TYPE assembly_module_or_assembly_group_component = SELECT
    (assembly_module,
    assembly_group_component);
END_TYPE; -- assembly_module_or_assembly_group_component

TYPE assembly_or_interconnect_design_object_select = SELECT
    (assembly_component,
    component_feature,
    interconnect_module_design_object_select);
END_TYPE; -- assembly_or_interconnect_design_object_select

TYPE axis_placement_or_cto2d = SELECT
    (axis_placement,
    cartesian_transformation_operator_2d);
END_TYPE; -- axis_placement_or_cto2d

TYPE axis_placement_or_datum_based_vector_orientation_select = SELECT
    (axis_placement,
    datum_based_vector_orientation);
END_TYPE; -- axis_placement_or_datum_based_vector_orientation_select

TYPE bare_die_component_or_packaged_component = SELECT
    (packaged_component,
    bare_die_component);
END_TYPE; -- bare_die_component_or_packaged_component

TYPE bare_die_terminal_or_packaged_part_terminal = SELECT

```

```

    (bare_die_terminal,
      packaged_part_terminal);
END_TYPE; -- bare_die_terminal_or_packaged_part_terminal

```

```

TYPE boolean_csg_operand = SELECT
  (csg_primitive,
    bound_volume_shape,
    boolean_csg_result,
    primitive_2d);
END_TYPE; -- boolean_csg_operand

```

```

TYPE boolean_operator = ENUMERATION OF
  (difference,
    intersection,
    union);
END_TYPE; -- boolean_operator

```

```

TYPE boolean_property = BOOLEAN;
END_TYPE; -- boolean_property

```

```

TYPE bus_structural_definition_or_node = SELECT
  (functional_unit_network_node_definition,
    bus_structural_definition);
END_TYPE; -- bus_structural_definition_or_node

```

```

TYPE centre_type = ENUMERATION OF
  (centred_on_arc,
    centred_on_radial);
END_TYPE; -- centre_type

```

```

TYPE component_based_message_type = ENUMERATION OF
  (component_reference_designation,
    terminal_designation);
END_TYPE; -- component_based_message_type

```

```

TYPE compound_characteristic_definition = SELECT
    (list_characteristic,
        set_characteristic);
END_TYPE; -- compound_characteristic_definition

```

```

TYPE component_feature_or_stratum_surface = SELECT
    (stratum_surface,
        component_feature);
END_TYPE; -- component_feature_or_stratum_surface

```

```

TYPE component_or_component_feature = SELECT
    (assembly_component,
        component_external_reference,
        component_feature,
        component_feature_external_reference);
END_TYPE; -- component_or_component_feature

```

```

TYPE component_or_feature = SELECT
    (assembly_component,
        assembly_joint,
        component_feature);
END_TYPE; -- component_or_feature

```

```

TYPE conductive_interconnect_terminal_or_interconnect_component_join_terminal = SELECT
    (interconnect_component_interface_terminal,
        interconnect_component_join_terminal,
        movable_packaged_component_join_terminal);
END_TYPE; -- conductive_interconnect_terminal_or_interconnect_component_join_terminal

```

```

TYPE constraining_element_definitions = SELECT
    (tolerance_zone_definition,
        profile_related_positional_boundary_definition,
        positional_boundary_member_definition,

```

```
        conical_tolerance_zone_boundary_diametrical_size_characteristic);  
END_TYPE; -- constraining_element_definitions
```

```
TYPE correlated_or_independent = SELECT  
    (coordinated_characteristic,  
  
        characteristic);  
END_TYPE; -- correlated_or_independent
```

```
TYPE curve_font_or_scaled_curve_font_select = SELECT  
    (curve_style_font_and_scaling,  
  
        curve_style_font_select);  
END_TYPE; -- curve_font_or_scaled_curve_font_select
```

```
TYPE curve_style_font_select = SELECT  
    (solid_curve,  
  
        externally_defined_curve_font,  
  
        curve_style_font);  
END_TYPE; -- curve_style_font_select
```

```
TYPE date_or_date_and_time = SELECT  
    (date_and_time,  
  
        date);  
END_TYPE; -- date_or_date_and_time
```

```
TYPE degree_or_radian = ENUMERATION OF  
    (degree,  
  
        radian);  
END_TYPE; -- degree_or_radian
```

```
TYPE degree_of_freedom = ENUMERATION OF  
    (x,  
  
        y,  
  
        z,  
  
        u,
```

```

        v,
        w);
END_TYPE; -- degree_of_freedom

TYPE descriptive_or_geometric_orientation = SELECT
    (measure_orientation,
     direction);
END_TYPE; -- descriptive_or_geometric_orientation

TYPE design_composition_path_or_root = SELECT
    (design_composition_path,
     functional_unit_network_definition);
END_TYPE; -- design_composition_path_or_root

TYPE design_intent_modification_3d_shape_select = SELECT
    (design_intent_modification_non_planar_2d_shape,
     design_intent_modification_3d_shape);
END_TYPE; -- design_intent_modification_3d_shape_select

TYPE design_intent_modification_planar_shape_select = SELECT
    (design_intent_modification_styled_area_representation,
     design_intent_modification_planar_shape);
END_TYPE; -- design_intent_modification_planar_shape_select

TYPE design_intent_modification_or_material_removal_component = SELECT
    (design_intent_modification,
     material_removal_laminate_component);
END_TYPE; -- design_intent_modification_or_material_removal_component

TYPE design_object_management_relationship_or_description = SELECT
    (ee_text,
     design_object_management_relationship);
END_TYPE; -- design_object_management_relationship_or_description

```

```
TYPE dimension_qualifiers = ENUMERATION OF
    (free_state,
        average);
END_TYPE; -- dimension_qualifiers
```

```
TYPE dimensional_location_or_dimensional_size = SELECT
    (dimensional_size,
        dimensional_location);
END_TYPE; -- dimensional_location_or_dimensional_size
```

```
TYPE dimensional_types = ENUMERATION OF
    (angular,
        radius,
        arc_length,
        diameter,
        linear,
        controlled_radius);
END_TYPE; -- dimensional_types
```

```
TYPE document_or_organization = SELECT
    (ee_document,
        organization);
END_TYPE; -- document_or_organization
```

```
TYPE ee_name = label;
END_TYPE; -- ee_name
```

```
TYPE ee_product_definition_with_annotation_elements = SELECT
    (footprint_definition,
        padstack_definition,
        physical_unit,
        stratum,
        template_definition);
END_TYPE; -- ee_product_definition_with_annotation_elements
```



```
TYPE environment = ENUMERATION OF
  (end_user_application,
    manufacturing);
END_TYPE; -- environment
```

```
TYPE evaluated_characteristic_or_description = SELECT
  (evaluated_characteristic,
    ee_text);
END_TYPE; -- evaluated_characteristic_or_description
```

```
TYPE evaluation_result_value = ENUMERATION OF
  (pass,
    fail,
    undefined);
END_TYPE; -- evaluation_result_value
```

```
TYPE evaluation_type = ENUMERATION OF
  (not_evaluable,
    invalid,
    valid,
    to_be_evaluated);
END_TYPE; -- evaluation_type
```

```
TYPE extend_or_chord_2_extend_or_truncate_or_round = ENUMERATION OF
  (truncate,
    chord_2_extend,
    round,
    extend);
END_TYPE; -- extend_or_chord_2_extend_or_truncate_or_round
```

```
TYPE extend_or_truncate = ENUMERATION OF
  (truncate,
    extend);
END_TYPE; -- extend_or_truncate
```

```
TYPE fabricated_feature_select = SELECT
  (component_termination_passage_join_terminal,
    via_terminal,
    printed_component_join_terminal,
    conductive_interconnect_element_terminal,
    embedded_component_terminal,
    land_join_terminal,
    non_functional_land_join_terminal);
END_TYPE; -- fabricated_feature_select
```

```
TYPE fill_area_style_tile_shape_select = SELECT
  (externally_defined_tile,
    fill_area_style_tile_symbol_with_style,
    fill_area_style_tile_solid_region,
    fill_area_style_tile_curve_with_style);
END_TYPE; -- fill_area_style_tile_shape_select
```

```
TYPE fill_style_select = SELECT
  (solid_fill_style,
    fill_area_style_tile,
    externally_defined_tile_style,
    externally_defined_hatch_style,
    fill_area_style_hatching);
END_TYPE; -- fill_style_select
```

```
TYPE font_select = SELECT
  (registered_font,
    font_definition);
END_TYPE; -- font_select
```

```
TYPE geometric_or_dimension_select = SELECT
  (geometric_tolerance,
```

```

        dimensional_characteristic);
END_TYPE; -- geometric_or_dimension_select

```

```

TYPE geometric_representation_relationship_purpose = ENUMERATION OF
    (non_feature_shape_annotation);
END_TYPE; -- geometric_representation_relationship_purpose

```

```

TYPE geometric_status = SELECT
    (predefined_geometric_status,
        external_definition);
END_TYPE; -- geometric_status

```

```

TYPE geometric_tolerance_qualifiers = ENUMERATION OF
    (pitch_diameter,
        minor_diameter,
        each_element,
        major_diameter,
        each_radial_element,
        not_concave,
        not_convex,
        free_state);
END_TYPE; -- geometric_tolerance_qualifiers

```

```

TYPE horizontal_justification = ENUMERATION OF
    (right,
        centre,
        left);
END_TYPE; -- horizontal_justification

```

```

TYPE interconnect_module_design_object_category = ENUMERATION OF
    (interconnect_module_via,
        interconnect_module_inter_stratum_feature,

```

```
        interconnect_module_cutout,  
        interconnect_module_fill_area,  
        interconnect_module_stratum_feature);  
END_TYPE; -- interconnect_module_design_object_category
```

```
TYPE interconnect_module_design_object_select = SELECT  
    (via,  
     cutout,  
     component_termination_passage,  
     land,  
     conductor,  
     conductive_filled_area,  
     stratum_feature,  
     laminate_component,  
     inter_stratum_feature,  
     non_functional_land);  
END_TYPE; -- interconnect_module_design_object_select
```

```
TYPE template_select = SELECT  
    (stratum_feature_template,  
     material_addition_feature_template,  
     material_removal_feature_template);  
END_TYPE; -- template_select
```

```
TYPE terminal_or_inter_stratum_feature = SELECT  
    (inter_stratum_feature,  
     embedded_component_terminal,  
     interconnect_module_macro_component_join_terminal,  
     interconnect_module_terminal,  
     printed_component_join_terminal);  
END_TYPE; -- terminal_or_inter_stratum_feature
```

```

TYPE interconnect_or_assembly_module_select = SELECT
  (component_termination_passage_interface_terminal,
    printed_connector_component_interface_terminal,
    non_functional_land_interface_terminal,
    land_interface_terminal);
END_TYPE; -- interconnect_or_assembly_module_select

```

```

TYPE keepout_product_design_object_category = ENUMERATION OF
  (assembly_module_assembly_component,
    assembly_module_component_feature,
    assembly_ee_material,
    interconnect_ee_material,
    interconnect_module_assembly_component,
    interconnect_module_via,
    interconnect_module_inter_stratum_feature,
    interconnect_module_cutout,
    interconnect_module_fill_area,
    interconnect_module_laminate_component,
    interconnect_module_stratum_feature);
END_TYPE; -- keepout_product_design_object_category

```

```

TYPE keepout_purpose = SELECT
  (predefined_keepout_purpose,
    external_definition);
END_TYPE; -- keepout_purpose

```

```

TYPE label = STRING;
END_TYPE; -- label

```

```

TYPE land_template_terminal_class = ENUMERATION OF
  (surface_point,

```

```

        edge_curve,

        edge_point,

        surface_area);
END_TYPE; -- land_template_terminal_class

```

```

TYPE layer_position_type = ENUMERATION OF
    (secondary,

    all,

    external,

    primary,

    internal);
END_TYPE; -- layer_position_type

```

```

TYPE layer_purpose = ENUMERATION OF
    (other_signal,

    lands_only,

    power_or_ground);
END_TYPE; -- layer_purpose

```

```

TYPE lead_form_type = SELECT
    (predefined_lead_form,

    external_definition);
END_TYPE; -- lead_form_type

```

```

TYPE life_cycle_stage = ENUMERATION OF
    (conceptual_design,

    final_design,

    detailed_design,

    preliminary_design);
END_TYPE; -- life_cycle_stage

```

```

TYPE list_characteristic = LIST[1:?] OF characteristic;
END_TYPE; -- list_characteristic

```

```

TYPE locating_or_refinement_linear_profile_tolerance = SELECT
    (composite_linear_profile_locating_tolerance,
        composite_linear_profile_refinement_tolerance);
END_TYPE; -- locating_or_refinement_linear_profile_tolerance

```

```

TYPE locating_or_refinement_surface_profile_tolerance = SELECT
    (composite_surface_profile_locating_tolerance,
        composite_surface_profile_refinement_tolerance);
END_TYPE; -- locating_or_refinement_surface_profile_tolerance

```

```

TYPE locating_or_relating_position_tolerance = SELECT
    (composite_feature_locating_position_tolerance,
        composite_feature_relating_position_tolerance);
END_TYPE; -- locating_or_relating_position_tolerance

```

```

TYPE logical_property = LOGICAL;
END_TYPE; -- logical_property

```

```

TYPE markable_item = SELECT
    (assembly_module,
        bare_die,
        interconnect_module,
        package,
        pca,
        pcb,
        packaged_part);
END_TYPE; -- markable_item

```

```

TYPE marked_message_select = SELECT
    (configuration_controlled_message,
        ee_text);
END_TYPE; -- marked_message_select

```

```
TYPE material_condition = ENUMERATION OF
  (nominal_material_condition,
    minimum_material_condition,
    maximum_material_condition);
END_TYPE; -- material_condition
```

```
TYPE material_condition_property_names = ENUMERATION OF
  (maximum_material_principle,
    least_material_requirement);
END_TYPE; -- material_condition_property_names
```

```
TYPE material_conductivity_classification = ENUMERATION OF
  (conductive,
    resistive,
    semi_conductive,
    non_conductive);
END_TYPE; -- material_conductivity_classification
```

```
TYPE material_state_change_enumeration = ENUMERATION OF
  (material_addition,
    material_removal);
END_TYPE; -- material_state_change_enumeration
```

```
TYPE material_or_specification = SELECT
  (ee_requirement_occurrence,
    ee_material);
END_TYPE; -- material_or_specification
```

```
TYPE mating_connector_part = SELECT
  (ee_specification,
    printed_connector_template,
    packaged_connector);
END_TYPE; -- mating_connector_part
```



```
TYPE measure_orientation = ENUMERATION OF
    (reversed,
        normal);
END_TYPE; -- measure_orientation
```

```
TYPE message_content = STRING;
END_TYPE; -- message_content
```

```
TYPE message_source_select = SELECT
    (organization,
        ee_approval,
        ee_product_version);
END_TYPE; -- message_source_select
```

```
TYPE message_type = ENUMERATION OF
    (part_number,
        manufacturer_code,
        revision_identifier,
        approval_date);
END_TYPE; -- message_type
```

```
TYPE metal_or_dielectric = ENUMERATION OF
    (over_finished_conductor,
        over_finished_dielectric);
END_TYPE; -- metal_or_dielectric
```

```
TYPE model_parameter_type = ENUMERATION OF
    (string_property_type,
        boolean_property_type,
        logical_property_type,
        numeric_property_type,
        physical_property_type);
END_TYPE; -- model_parameter_type
```

```

TYPE month_representation = INTEGER;
END_TYPE; -- month_representation

```

```

TYPE mounting_technology_type = ENUMERATION OF
  (surface_mount,

    through_hole);
END_TYPE; -- mounting_technology_type

```

```

TYPE nomenclature = LIST[1:?] OF STRING;
END_TYPE; -- nomenclature

```

```

TYPE non_numeric_property_value = SELECT
  (string_property,

    boolean_property,

    logical_property);
END_TYPE; -- non_numeric_property_value

```

```

TYPE numeric_or_non_numeric_select = SELECT
  (external_definition,

    compound_characteristic_definition,

    data_element,

    non_numeric_property_value);
END_TYPE; -- numeric_or_non_numeric_select

```

```

TYPE offset_types = ENUMERATION OF
  (towards_material,

    away_from_material);
END_TYPE; -- offset_types

```

```

TYPE orientation_select = SELECT
  (datum_axis_related_orientation,

    viewing_plane_based_orientation,

    datum_based_vector_orientation);
END_TYPE; -- orientation_select

```

```

TYPE orientation_type = ENUMERATION OF
    (parallel,
        undefined,
        perpendicular);
END_TYPE; -- orientation_type

```

```

TYPE package_or_package_external_reference = SELECT
    (package,
        package_external_reference);
END_TYPE; -- package_or_package_external_reference

```

```

TYPE package_terminal_or_package_terminal_external_reference = SELECT
    (package_terminal,
        package_terminal_external_reference);
END_TYPE; -- package_terminal_or_package_terminal_external_reference

```

```

TYPE packaged_or_printed_terminal = SELECT
    (packaged_part_terminal,
        printed_part_template_terminal);
END_TYPE; -- packaged_or_printed_terminal

```

```

TYPE part_feature_or_template_feature = SELECT
    (land_template_terminal,
        component_termination_passage_template_terminal,
        part_feature,
        printed_part_template_terminal,
        part_terminal_external_reference,
        via_template_terminal);
END_TYPE; -- part_feature_or_template_feature

```

```

TYPE part_template_or_physical_unit_2d_shape_select = SELECT
    (part_template_planar_shape,
        physical_unit_planar_shape);
END_TYPE; -- part_template_or_physical_unit_2d_shape_select

```

```

TYPE part_template_or_physical_unit_3d_shape_select = SELECT
    (part_template_3d_shape,

    part_template_non_planar_2d_shape,

    physical_unit_3d_shape);
END_TYPE; -- part_template_or_physical_unit_3d_shape_select

```

```

TYPE part_template_or_physical_unit_shape_select = SELECT
    (part_template_or_physical_unit_2d_shape_select,

    part_template_or_physical_unit_3d_shape_select);
END_TYPE; -- part_template_or_physical_unit_shape_select

```

```

TYPE pca_interface_select = SELECT
    (interconnect_module_component_terminal,

    component_interface_terminal,

    assembly_module_component_terminal);
END_TYPE; -- pca_interface_select

```

```

TYPE pca_product_design_object_category = ENUMERATION OF
    (assembly_module_assembly_component,

    assembly_module_assembly_module,

    assembly_module_assembly_module_component,

    assembly_module_assembly_module_component_terminal,

    assembly_module_assembly_module_macro_component,

    assembly_module_assembly_module_macro_component_join_terminal,

    assembly_module_bare_die_component,

    assembly_module_bare_die_component_terminal,

    assembly_module_cable_component,

    assembly_module_component_feature,

    assembly_module_component_mounting_feature,

    assembly_module_interconnect_component_interface_terminal,

```

assembly_module_interconnect_component_join_terminal,
assembly_module_interconnect_module_component,
assembly_module_interconnect_module_component_stratum_based_terminal,
assembly_module_interconnect_module_component_surface_feature,
assembly_module_interconnect_module_component_terminal,
assembly_module_interconnect_module_macro_component,
assembly_module_interconnect_module_macro_component_join_terminal,
assembly_module_interface_component,
assembly_module_minimally_defined_component_terminal,
assembly_module_movable_packaged_component_join_terminal,
assembly_module_packaged_component,
assembly_module_packaged_component_join_terminal,
assembly_module_packaged_connector_component,
assembly_module_packaged_connector_component_interface_terminal,
assembly_module_physical_component,
assembly_module_routed_interconnect_component,
assembly_module_routed_physical_component,
interconnect_module_assembly_component,
interconnect_module_assembly_module,
interconnect_module_bonded_conductive_base_blind_via,
interconnect_module_buried_via,
interconnect_module_component_termination_passage,
interconnect_module_component_termination_passage_and_contact_size_dependent_land,
interconnect_module_component_termination_passage_and_contact_size_dependent_non_func
interconnect_module_component_termination_passage_dependent_land,
interconnect_module_component_termination_passage_dependent_non_functional_land,

interconnect_module_component_termination_passage_interface_terminal,
interconnect_module_component_termination_passage_join_terminal,
interconnect_module_conductive_interconnect_element_terminal,
interconnect_module_conductive_interconnect_element_with_pre_defined_transitions,
interconnect_module_conductive_interconnect_element_with_user_defined_single_transitions,
interconnect_module_connected_area_component,
interconnect_module_contact_size_dependent_land,
interconnect_module_contact_size_dependent_non_functional_land,
interconnect_module_cutout,
interconnect_module_cutout_edge_segment,
interconnect_module_dielectric_material_passage,
interconnect_module_electrical_isolation_laminate_component,
interconnect_module_embedded_component_terminal,
interconnect_module_fiducial,
interconnect_module_fill_area,
interconnect_module_filled_area_material_removal_laminate_component,
interconnect_module_inter_stratum_feature,
interconnect_module_interconnect_module_edge,
interconnect_module_interconnect_module_edge_segment,
interconnect_module_interface_access_material_removal_laminate_component,
interconnect_module_interface_access_stratum_feature_template_component,
interconnect_module_interfacial_connection,
interconnect_module_internal_probe_access_area,
interconnect_module_join_two_physical_connectivity_definition_supporting_inter_stratum,
interconnect_module_join_two_physical_connectivity_definition_supporting_printed_component,
interconnect_module_laminate_component,

interconnect_module_laminate_text_component,
interconnect_module_laminate_text_string_component,
interconnect_module_land,
interconnect_module_land_interface_terminal,
interconnect_module_land_join_terminal,
interconnect_module_material_removal_laminate_component,
interconnect_module_multi_layer_material_removal_laminate_component,
interconnect_module_multi_layer_stratum_feature_template_component,
interconnect_module_non_conductive_base_blind_via,
interconnect_module_non_functional_land,
interconnect_module_non_functional_land_interface_terminal,
interconnect_module_non_functional_land_join_terminal,
interconnect_module_partially_plated_cutout,
interconnect_module_partially_plated_interconnect_module_edge,
interconnect_module_physical_connectivity_interrupting_cutout,
interconnect_module_physical_laminate_component,
interconnect_module_physical_network_supporting_inter_stratum_feature,
interconnect_module_plated_conductive_base_blind_via,
interconnect_module_plated_cutout,
interconnect_module_plated_cutout_edge_segment,
interconnect_module_plated_interconnect_module_edge,
interconnect_module_plated_interconnect_module_edge_segment,
interconnect_module_plated_passage,
interconnect_module_plated_passage_or_unsupported_passage,
interconnect_module_primary_stratum_indicator_symbol,
interconnect_module_printed_component_join_terminal,

```

interconnect_module_printed_connector_component_interface_terminal,
interconnect_module_probe_access_area,
interconnect_module_special_symbol_laminate_component,
interconnect_module_stratum_feature,
interconnect_module_stratum_feature_template_component,
interconnect_module_thermal_isolation_laminate_component,
interconnect_module_unrouted_conductive_interconnect_element,
interconnect_module_unsupported_passage,
interconnect_module_unsupported_passage_dependent_non_functional_land,
interconnect_module_via,
interconnect_module_via_and_contact_size_dependent_land,
interconnect_module_via_and_contact_size_dependent_non_functional_land,
interconnect_module_via_dependent_land,
interconnect_module_via_dependent_non_functional_land,
interconnect_module_via_terminal);
END_TYPE; -- pca_product_design_object_category

```

```

TYPE person_organization = SELECT
  (person_and_organization,
    organization);
END_TYPE; -- person_organization

```

```

TYPE physical_unit_or_part_template = SELECT
  (part_template,
    physical_unit);
END_TYPE; -- physical_unit_or_part_template

```

```

TYPE physical_unit_or_part_template_or_fp_or_pd = SELECT
  (part_template,
    physical_unit,

```



```

        footprint_definition,
        padstack_definition);
END_TYPE; -- physical_unit_or_part_template_or_fp_or_pd

```

```

TYPE plain_or_structured_text = SELECT
    (structured_text,
    ee_text);
END_TYPE; -- plain_or_structured_text

```

```

TYPE planar_purpose = SELECT
    (predefined_planar_purpose,
    external_definition);
END_TYPE; -- planar_purpose

```

```

TYPE positive_integer = INTEGER;
END_TYPE; -- positive_integer

```

```

TYPE positive_ratio_measure = REAL;
END_TYPE; -- positive_ratio_measure

```

```

TYPE precedence_type = ENUMERATION OF
    (secondary,
    tertiary,
    primary);
END_TYPE; -- precedence_type

```

```

TYPE predefined_geometric_status = ENUMERATION OF
    (undefined,
    standard_datum_condition,
    perpendicular_primary_datum_axis,
    perpendicular_primary_datum_plane);
END_TYPE; -- predefined_geometric_status

```

```

TYPE predefined_lead_form = ENUMERATION OF

```

```
(gull_wing,  
    j_lead,  
    reversed_j_lead,  
    integral_terminal,  
    undefined,  
    ball,  
    straight);  
END_TYPE; -- predefined_lead_form
```

```
TYPE predefined_keepout_purpose = ENUMERATION OF  
    (thermal,  
        generic_clearance,  
        shock,  
        vibration,  
        electromagnetic_compatibility);  
END_TYPE; -- predefined_keepout_purpose
```

```
TYPE predefined_planar_purpose = ENUMERATION OF  
    (analysis_input,  
        analysis_output,  
        shock_analysis_input,  
        shock_analysis_output,  
        assembly_symbol,  
        design,  
        design_profile,  
        vibration_analysis_input,  
        vibration_analysis_output,  
        electromagnetic_compatibility_analysis_input,  
        electromagnetic_compatibility_analysis_output,
```

```

    physical_extent,
    thermal_analysis_input,
    thermal_analysis_output);
END_TYPE; -- predefined_planar_purpose

```

```

TYPE predefined_3d_purpose = ENUMERATION OF
    (analysis_input,
    analysis_output,
    shock_analysis_input,
    shock_analysis_output,
    design,
    vibration_analysis_input,
    vibration_analysis_output,
    electromagnetic_compatibility_analysis_input,
    electromagnetic_compatibility_analysis_output,
    thermal_analysis_input,
    thermal_analysis_output);
END_TYPE; -- predefined_3d_purpose

```

```

TYPE printed_part_template_or_printed_part_cross_section_template = SELECT
    (printed_part_template,
    printed_part_cross_section_template);
END_TYPE; -- printed_part_template_or_printed_part_cross_section_template

```

```

TYPE printed_part_template_terminal_class = ENUMERATION OF
    (curve_edge_segment,
    surface_point,
    area_edge_segment,
    surface_area);
END_TYPE; -- printed_part_template_terminal_class

```

```
TYPE product_object_select = SELECT
    (ee_product_configuration,
      ee_product_definition,
      ee_product_version,
      ee_product,
      part_feature,
      physical_connectivity_element,
      physical_connectivity_definition,
      restriction_basis_item);
END_TYPE; -- product_object_select
```

```
TYPE range_class = ENUMERATION OF
    (closed,
      lower_open,
      upper_open,
      open);
END_TYPE; -- range_class
```

```
TYPE relationship_operation = ENUMERATION OF
    (and_operation,
      derived_from,
      or_operation);
END_TYPE; -- relationship_operation
```

```
TYPE requirement_allocation_select = SELECT
    (assembly_requirement_allocation,
      requirement_allocation,
      functional_unit_requirement_allocation);
END_TYPE; -- requirement_allocation_select
```

```
TYPE restrained_item_select = SELECT
    (datum_feature_usage_in_datum_relationship,
```

```

    physical_feature_or_part_template);
END_TYPE; -- restrained_item_select

```

```

TYPE restriction_basis_item = SELECT
    (stratum_concept,

    ee_requirement_occurrence,

    part_feature,

    stratum,

    component_or_component_feature);
END_TYPE; -- restriction_basis_item

```

```

TYPE scalar_or_coordinated_characteristics = SELECT
    (coordinated_characteristic,

    characteristic);
END_TYPE; -- scalar_or_coordinated_characteristics

```

```

TYPE seating_plane_intersection_type = ENUMERATION OF
    (surface_intersection,

    through_intersection,

    does_not_intersect);
END_TYPE; -- seating_plane_intersection_type

```

```

TYPE set_characteristic = SET[1:?] OF characteristic;
END_TYPE; -- set_characteristic

```

```

TYPE shape_location = ENUMERATION OF
    (above,

    below,

    congruent);
END_TYPE; -- shape_location

```

```

TYPE shieldable_item = SELECT
    (assembly_component,

    component_feature,

```

```
    physical_connectivity_definition);  
END_TYPE; -- shieldable_item
```

```
TYPE si_prefix = ENUMERATION OF  
    (exa,  
  
    peta,  
  
    tera,  
  
    giga,  
  
    mega,  
  
    kilo,  
  
    hecto,  
  
    deca,  
  
    deci,  
  
    centi,  
  
    milli,  
  
    micro,  
  
    nano,  
  
    pico,  
  
    femto,  
  
    atto);  
END_TYPE; -- si_prefix
```

```
TYPE signal_flow_direction = ENUMERATION OF  
    (unknown_direction,  
  
    not_applicable,  
  
    output_direction,  
  
    input_direction,  
  
    bidirectional);  
END_TYPE; -- signal_flow_direction
```

```

TYPE square_or_round = ENUMERATION OF
    (square,
        round);
END_TYPE; -- square_or_round

```

```

TYPE spacing_type = ENUMERATION OF
    (nearest_boundary,
        centroid,
        furthest_boundary);
END_TYPE; -- spacing_type

```

```

TYPE specification_product_object_select = SELECT
    (ee_product_configuration,
        ee_product_definition,
        ee_product_version,
        ee_product,
        part_feature,
        physical_connectivity_element,
        physical_connectivity_definition,
        stratum_concept,
        stratum,
        component_or_component_feature);
END_TYPE; -- specification_product_object_select

```

```

TYPE stackup_select = SELECT
    (passage_technology,
        via_template,
        component_termination_passage_template);
END_TYPE; -- stackup_select

```

```

TYPE standard_document_identification_string = STRING;
END_TYPE; -- standard_document_identification_string

```

```

TYPE standard_or_reference_document = SELECT
    (ee_document,

        standard_document_identification_string);
END_TYPE; -- standard_or_reference_document

```

```

TYPE status = ENUMERATION OF
    (disapproved,

        withdrawn,

        not_yet_approved,

        approved);
END_TYPE; -- status

```

```

TYPE stiffness_class = ENUMERATION OF
    (fluid_like,

        conformal_coat,

        stiff_laminate);
END_TYPE; -- stiffness_class

```

```

TYPE stratum_concept = SELECT
    (laminate_component,

        inter_stratum_feature,

        stratum_feature,

        stratum,

        stratum_surface);
END_TYPE; -- stratum_concept

```

```

TYPE stratum_feature_3d_shape_select = SELECT
    (stratum_feature_non_planar_2d_shape,

        stratum_feature_3d_shape);
END_TYPE; -- stratum_feature_3d_shape_select

```

```

TYPE stratum_feature_planar_representation_select = SELECT
    (stratum_feature_styled_area_representation,

```



```

    stratum_feature_planar_shape);
END_TYPE; -- stratum_feature_planar_representation_select

```

```

TYPE stratum_surface_designation = ENUMERATION OF
    (primary_surface,

    secondary_surface);
END_TYPE; -- stratum_surface_designation

```

```

TYPE stratum_surface_shape_select = SELECT
    (stratum_surface_shape,

    stratum_average_surface_shape);
END_TYPE; -- stratum_surface_shape_select

```

```

TYPE string_property = STRING;
END_TYPE; -- string_property

```

```

TYPE styled_area_representation = SELECT
    (fill_area_occurrence,

    styled_curve_occurrence);
END_TYPE; -- styled_area_representation

```

```

TYPE termination_or_junction = SELECT
    (component_terminal,

    topological_junction);
END_TYPE; -- termination_or_junction

```

```

TYPE text = STRING;
END_TYPE; -- text

```

```

TYPE thermal_isolation_removal_structure_or_template = SELECT
    (thermal_isolation_removal_structure,

    thermal_isolation_removal_template);
END_TYPE; -- thermal_isolation_removal_structure_or_template

```

```

TYPE three_d_purpose = SELECT

```

```

        (predefined_3d_purpose,
        external_definition);
END_TYPE; -- three_d_purpose

```

```

TYPE toleranceable_shape_element = SELECT
    (centre_axis,
    centre_plane,
    centre_point,
    tangent_plane,
    physical_feature_or_part_template);
END_TYPE; -- toleranceable_shape_element

```

```

TYPE trace_junction_or_via = SELECT
    (inter_stratum_join_relationship,
    explicitly_located_layer_connection_point);
END_TYPE; -- trace_junction_or_via

```

```

TYPE trimmed_or_bounding_curve = SELECT
    (bounding_curve,
    trimmed_line);
END_TYPE; -- trimmed_or_bounding_curve

```

```

TYPE true_profile_representations = SELECT
    (dimension_set,
    ee_document,
    ee_rule_function,
    bound_volume_shape);
END_TYPE; -- true_profile_representations

```

```

TYPE two_or_three_dimensional = ENUMERATION OF
    (three_dimensional,
    two_dimensional);
END_TYPE; -- two_or_three_dimensional

```

```
TYPE usage_concept = SELECT
  (land_template_terminal,
    component_termination_passage_template_terminal,
    via_template_terminal,
    non_feature_shape_element,
    part_feature,
    printed_part_template_terminal);
END_TYPE; -- usage_concept
```

```
TYPE usage_view_terminal = SELECT
  (assembly_module_terminal,
    bare_die_terminal,
    component_termination_passage_template_terminal,
    interconnect_module_terminal,
    land_template_terminal,
    package_terminal,
    printed_part_template_terminal,
    via_template_terminal,
    part_mounting_feature);
END_TYPE; -- usage_view_terminal
```

```
TYPE value_qualifier = SELECT
  (precision_qualifier,
    type_qualifier,
    uncertainty_qualifier);
END_TYPE; -- value_qualifier
```

```
TYPE vertical_justification = ENUMERATION OF
  (centre,
    top,
```

```

        bottom);
END_TYPE; -- vertical_justification

```

```

TYPE view_shape_select = SELECT
    (part_template_3d_shape,

    physical_unit_3d_shape,

    part_template_non_planar_2d_shape,

    part_template_planar_shape,

    physical_unit_planar_shape);
END_TYPE; -- view_shape_select

```

```

ENTITY absorbed_dose_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- absorbed_dose_data_element

```

```

ENTITY activity_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- activity_data_element

```

```

ENTITY add_design_object_management_relationship
    SUBTYPE OF (design_object_management_relationship);
    current_design_object : managed_design_object;
    UNIQUE
        UR1 : current_design_object,
current_design;
END_ENTITY; -- add_design_object_management_relationship

```

```

ENTITY adjacent_stratum_surface_embedded_component_surface_definition
    SUBTYPE OF (managed_design_object);
    name : STRING;
    precedent_item : component_feature_or_stratum_surface;
    subsequent_item : component_feature_or_stratum_surface;
    UNIQUE
        UR1 : name;
        UR2 : precedent_item,
subsequent_item;
    WHERE
        WR1 : TYPEOF(precedent_item) <> TYPEOF(subsequent_item);
        WR2 : NOT ('AP210_ARM.DATUM_FEATURE_USAGE_IN_DATUM_RELATIONSHIP' IN TYPEOF(SELF));
        WR3 : precedent_item <> subsequent_item;
        WR4 : NOT ('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(precedent_item)) OR (SIZEOF(['AP2

```

```

'AP210_ARM.PACKAGE_BODY_SURFACE'] * TYPEOF(precedent_item.definition)) = 1);
    WR5 : NOT ('AP210_ARM.COMPONENT_FEATURE' IN TYPEOF(subsequent_item)) OR (SIZEOF(['AP210_ARM.PACKAGE_BODY_SURFACE'] * TYPEOF(subsequent_item.definition)) = 1);
END_ENTITY; -- adjacent_stratum_surface_embedded_component_surface_definition

```

```

ENTITY adjacent_stratum_surface_definition
  SUBTYPE OF (managed_design_object);
    name : STRING;
    precedent_surface : stratum_surface;
    subsequent_surface : stratum_surface;
  UNIQUE
    UR1 : name;
    UR2 : precedent_surface,
subsequent_surface;
END_ENTITY; -- adjacent_stratum_surface_definition

```

```

ENTITY advanced_b_rep
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- advanced_b_rep

```

```

ENTITY aggregate_connectivity_requirement
  SUBTYPE OF (managed_design_object);
    connectivity_requirement : functional_unit_network_node_definition;
    design_definition_path : design_composition_path_or_root;
  UNIQUE
    UR1 : design_definition_path,
connectivity_requirement;
  WHERE
    WR1 : NOT ('AP210_ARM.FUNCTIONAL_UNIT_NETWORK_DEFINITION' IN TYPEOF(design_definition_path));
    WR2 : NOT ('AP210_ARM.DESIGN_COMPOSITION_PATH' IN TYPEOF(design_definition_path)) OR (design_definition_path = design_composition_path_or_root);
END_ENTITY; -- aggregate_connectivity_requirement

```

```

ENTITY altered_package
  SUBTYPE OF (package);
    of_geometric_status : geometric_status;
    modified_terminal_separation : OPTIONAL length_data_element;
    package_to_be_altered : package;
END_ENTITY; -- altered_package

```

```

ENTITY altered_package_terminal
  ABSTRACT SUPERTYPE OF ((shape_formed_terminal ANDOR
surface_prepped_terminal) ANDOR
length_trimmed_terminal)
  SUBTYPE OF (package_terminal);
    terminal_to_be_altered : package_terminal;

```

```

        SELF\part_feature.associated_definition : altered_package;
END_ENTITY; -- altered_package_terminal

```

```

ENTITY altered_packaged_part
  SUBTYPE OF (packaged_part);
  SELF\packaged_part.used_package : altered_package;
  base_packaged_part              : packaged_part;
END_ENTITY; -- altered_packaged_part

```

```

ENTITY alternate_product
  SUBTYPE OF (alternate_select_product);
  SELF\alternate_select_product.defined_alternate_product : SET[1:1] OF ee_product;
END_ENTITY; -- alternate_product

```

```

ENTITY alternate_select_product
  ABSTRACT SUPERTYPE OF (ONEOF (assembly_alternate_product,
test_select_product,
alternate_product))
  SUBTYPE OF (managed_design_object);
  defined_alternate_product : SET[1:?] OF ee_product;
  defining_base_product    : ee_product;
  authorizing_agreement    : OPTIONAL contract;
END_ENTITY; -- alternate_select_product

```

```

ENTITY amount_of_substance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- amount_of_substance_data_element

```

```

ENTITY analytical_model
  SUBTYPE OF (managed_design_object);
  name                : ee_name;
  representation_language : language_reference_manual;
  reference_document  : ee_document;
  parameter           : SET[1:?] OF model_parameter;
  source_code         : ee_specification;
  associated_definition : analytical_model_definition;
  INVERSE
    access_mechanism : SET[1:?] OF analytical_model_port FOR accessed_analytical_model;
  UNIQUE
    UR1 : name;
END_ENTITY; -- analytical_model

```

```

ENTITY analytical_model_definition
  SUBTYPE OF (ee_product_definition);

```

```

    INVERSE
        associated_model_occurrence : analytical_model FOR associated_definition;
END_ENTITY; -- analytical_model_definition

```

```

ENTITY analytical_model_make_from_relationship
    SUBTYPE OF (managed_design_object);
    reusable_model : analytical_model;
    resultant_model : analytical_model;
    UNIQUE
        UR1 : reusable_model,
resultant_model;
    WHERE
        WR1 : reusable_model :<>: resultant_model;
END_ENTITY; -- analytical_model_make_from_relationship

```

```

ENTITY analytical_model_port
    SUPERTYPE OF (analytical_model_vector_port ANDOR
digital_analytical_model_port)
    SUBTYPE OF (managed_design_object);
    accessed_analytical_model : analytical_model;
    port_name : ee_name;
    unit_of_measure : OPTIONAL STRING;
    port_type : model_parameter_type;
    distributed_properties : BOOLEAN;
    nominal_signal_flow_direction : OPTIONAL signal_flow_direction;
    UNIQUE
        UR1 : port_name,
accessed_analytical_model;
END_ENTITY; -- analytical_model_port

```

```

ENTITY analytical_model_port_assignment
    SUBTYPE OF (managed_design_object);
    assigned_functional_unit_terminal : functional_unit_usage_view_terminal_definition;
    assigned_analytical_model_port : analytical_model_port;
    WHERE
        WR1 : NOT (('AP210_ARM.ANALYTICAL_MODEL_SCALAR_PORT' IN TYPEOF(assigned_analytical_m
END_ENTITY; -- analytical_model_port_assignment

```

```

ENTITY analytical_model_port_make_from_relationship
    SUBTYPE OF (managed_design_object);
    associated_make_from : analytical_model_make_from_relationship;
    reusable_feature : analytical_model_port;
    resultant_feature : analytical_model_port;
    UNIQUE
        UR1 : reusable_feature,
resultant_feature;
    WHERE

```

```

        WR1 : reusable_feature :<>: resultant_feature;
        WR2 : reusable_feature.accessed_analytical_model :<>: resultant_feature.accessed_ana
END_ENTITY; -- analytical_model_port_make_from_relationship

```

```

ENTITY analytical_model_scalar_port
  SUBTYPE OF (analytical_model_vector_port);
  WHERE
    WR1 : SELF\analytical_model_vector_port.size = 1;
END_ENTITY; -- analytical_model_scalar_port

```

```

ENTITY analytical_model_vector_port
  SUPERTYPE OF (analytical_model_scalar_port)
  SUBTYPE OF (analytical_model_port);
  size : positive_integer;
END_ENTITY; -- analytical_model_vector_port

```

```

ENTITY analytical_representation
  SUBTYPE OF (managed_design_object);
  model_parameter_assignment : SET[1:?] OF parameter_assignment;
  model_representation : analytical_model;
  analytical_representation_name : ee_name;
  UNIQUE
  UR1 : analytical_representation_name;
END_ENTITY; -- analytical_representation

```

```

ENTITY angle_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- angle_data_element

```

```

ENTITY angular_dimension_with_direction_vector
  SUBTYPE OF (dimensional_location);
  measurement_orientation : axis_placement_or_datum_based_vector_orientation_select;
  UNIQUE
  UR1 : SELF\dimensional_location.located_elements,
measurement_orientation;
END_ENTITY; -- angular_dimension_with_direction_vector

```

```

ENTITY angular_size_dimension
  SUBTYPE OF (dimensional_size);
  half_angle : BOOLEAN;
  major_angle : angle_relator;
END_ENTITY; -- angular_size_dimension

```



```

ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- angularity_tolerance

```

```

ENTITY apex
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- apex

```

```

ENTITY area_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- area_data_element

```

```

ENTITY array_placement_group
  ABSTRACT SUPERTYPE OF (ONEOF (laminate_text_string_component,
linear_array_placement_group_component,
rectangular_array_placement_group_component))
  SUBTYPE OF (assembly_group_component);
END_ENTITY; -- array_placement_group

```

```

ENTITY assembled_with_bonding
  SUBTYPE OF (assembly_joint);
  default_bond_definition : assembly_bond_definition;
  INVERSE
    bond_assembly_position : SET[0:1] OF bond_assembly_position FOR bond;
  WHERE
    WR1 : SELF\assembly_joint.assembly_features_1.definition :=: default_bond_definition
    WR2 : SELF\assembly_joint.assembly_features_2.definition :=: default_bond_definition
END_ENTITY; -- assembled_with_bonding

```

```

ENTITY assembled_with_fasteners
  SUBTYPE OF (assembly_joint);
  SELF\assembly_joint.associated_assembly : assembly_group_component;
END_ENTITY; -- assembled_with_fasteners

```

```

ENTITY assembly_alternate_product
  SUBTYPE OF (alternate_select_product);
  basis_context : physical_unit_design_view;
END_ENTITY; -- assembly_alternate_product

```

```

ENTITY assembly_bond_definition
  SUBTYPE OF (managed_design_object);

```

```

        analytical_assembly_bond_representation : OPTIONAL SET[1:?] OF analytical_representation;
        associated_definition                    : externally_defined_ee_product_definition;
        bond_category                          : external_definition;
        bond_material                          : ee_material;
        bonded_feature_1                       : part_feature;
        bonded_feature_2                       : part_feature;
        name                                    : ee_name;
    INVERSE
        default_shape : SET[0:1] OF default_assembly_bond_shape FOR characterized_bond_definition;
    UNIQUE
        UR1 : name;
    WHERE
        WR1 : SIZEOF(QUERY(prpc <* USEDIN(associated_definition\ee_product_definition.version
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | (prpc.name = 'technology specification
END_ENTITY; -- assembly_bond_definition

```

```

ENTITY assembly_component
    ABSTRACT SUPERTYPE OF (((ONEOF (physical_component,
printed_component,
laminated_component) ANDOR
shield) ANDOR
thermal_component) ANDOR
assembly_group_component)
    SUBTYPE OF (managed_design_object);
    usage_definition : OPTIONAL physical_unit_or_part_template_or_fp_or_pd;
    INVERSE
        assembly : SET[1:?] OF assembly_composition_relationship FOR composition;
        component_2d_shape : SET[0:1] OF assembly_component_2d_shape FOR shape_characterized_component;
        component_3d_shape : SET[0:1] OF assembly_component_3d_shape FOR shape_characterized_component;
        component_non_planar_shape : SET[0:1] OF assembly_component_non_planar_shape FOR shape_characterized_component;
    WHERE
        WR1 : NOT (EXISTS(usage_definition)) OR (('AP210_ARM.FOOTPRINT_OCCURRENCE' IN TYPEOF(usage_definition));
END_ENTITY; -- assembly_component

```

```

ENTITY assembly_component_2d_shape
    SUBTYPE OF (planar_projected_shape);
    shape_characterized_component : SET[1:?] OF assembly_component;
END_ENTITY; -- assembly_component_2d_shape

```

```

ENTITY assembly_component_3d_shape
    SUBTYPE OF (feature_shape_definition);
    shape_characterized_component : SET[1:?] OF assembly_component;
END_ENTITY; -- assembly_component_3d_shape

```

```

ENTITY assembly_component_non_planar_shape
    SUBTYPE OF (manifold_subsurface_shape,
feature_shape_definition);

```

```

    shape_characterized_component : SET[1:?] OF assembly_component;
END_ENTITY; -- assembly_component_non_planar_shape

```

```

ENTITY assembly_component_placement_link
  SUBTYPE OF (managed_design_object);
  precedent_element : assembly_component;
  subsequent_element : assembly_component;
  INVERSE
    referenced_by : linear_array_placement_group_component FOR element;
END_ENTITY; -- assembly_component_placement_link

```

```

ENTITY assembly_composition_relationship
  SUPERTYPE OF (ONEOF (next_higher_assembly_relationship))
  SUBTYPE OF (managed_design_object);
  assembly          : physical_unit_design_view;
  component          : assembly_component;
  security_code      : OPTIONAL security_classification;
  assembly_bonding_agent : OPTIONAL SET[1:?] OF ee_material;
  UNIQUE
    UR1 : assembly,
component;
END_ENTITY; -- assembly_composition_relationship

```

```

ENTITY assembly_group_component
  SUPERTYPE OF (interfaced_group_component ANDOR
array_placement_group ANDOR
(ONEOF (footprint_occurrence,
padstack_occurrence)))
  SUBTYPE OF (assembly_component);
  INVERSE
    components : SET[1:?] OF sub_assembly_relationship FOR relating_component;
END_ENTITY; -- assembly_group_component

```

```

ENTITY assembly_joint
  SUPERTYPE OF (assembled_with_fasteners ANDOR
assembled_with_bonding)
  SUBTYPE OF (managed_design_object);
  assembly_features_1 : assembled_feature_select;
  assembly_features_2 : assembled_feature_select;
  associated_assembly : assembly_module_or_assembly_group_component;
  description         : OPTIONAL STRING;
  name                : STRING;
  UNIQUE
    UR1 : name,
associated_assembly;
  WHERE
    WR1 : assembly_features_1 :<>: assembly_features_2;

```

```

      WR2 : ('AP210_ARM.ASSEMBLED_WITH_FASTENERS' IN TYPEOF(SELF)) OR (('AP210_ARM.COMPONE
      WR3 : ('AP210_ARM.ASSEMBLED_WITH_FASTENERS' IN TYPEOF(SELF)) OR (('AP210_ARM.COMPONE
END_ENTITY; -- assembly_joint

```

```
ENTITY assembly_material_composition_relationship
  SUBTYPE OF (managed_design_object);
    quantity          : OPTIONAL data_element;
    assembly           : physical_unit_design_view;
    component_material : material_definition;
    security_code       : OPTIONAL security_classification;
    as_required        : LOGICAL;
END_ENTITY; -- assembly_material_composition_relationship
```

```
ENTITY assembly_module
  SUPERTYPE OF (ONEOF (pca,
reference_packaged_part_assembly_implementation))
  SUBTYPE OF (physical_unit_design_view);
END_ENTITY; -- assembly_module
```

```

ENTITY assembly_module_component
  SUPERTYPE OF (assembly_module_macro_component)
  SUBTYPE OF (physical_component);
    SELF\assembly_component.usage_definition : assembly_module_usage_view;
  INVERSE
    access_mechanisms : SET[2:?] OF assembly_module_component_terminal FOR associated_co
END_ENTITY; -- assembly_module_component

```

```
ENTITY assembly_module_component_terminal
  SUBTYPE OF (physical_component_terminal);
    SELF\component_feature.definition      : assembly_module_terminal;
    SELF\component_feature.associated_component : assembly_module_component;
END_ENTITY; -- assembly_module_component_terminal
```

```
ENTITY assembly_module_interface_terminal
  SUBTYPE OF (assembly_module_terminal);
END_ENTITY; -- assembly_module_interface_terminal
```

```

ENTITY assembly_module_join_terminal
  SUBTYPE OF (assembly_module_terminal);
  INVERSE
    defining_terminal : component_feature_to_physical_usage_view_assignment FOR defined_p
  WHERE
    WR1 : 'AP210_ARM.INTERCONNECT_MODULE_JOIN_TERMINAL' IN TYPEOF(defining_terminal.assi
    WR2 : 'AP210_ARM.COMPONENT_TERMINAL_TO_ASSEMBLY_MODULE_JOIN_TERMINAL_ASSIGNMENT' IN
END_ENTITY; -- assembly_module_join_terminal

```

```

ENTITY assembly_module_macro_component
  SUBTYPE OF (assembly_module_component);
    design_definition : assembly_module;
  INVERSE
    join_access_mechanisms : SET[1:?] OF assembly_module_macro_component_join_terminal FOR
  WHERE
    WR1 : SIZEOF(QUERY(ammcjt <* join_access_mechanisms | ammcjt\component_feature.definition
  END_ENTITY; -- assembly_module_macro_component

```

```

ENTITY assembly_module_macro_component_join_terminal
  SUBTYPE OF (assembly_module_component_terminal);
    SELF\component_feature.definition : assembly_module_join_terminal;
    SELF\component_feature.associated_component : assembly_module_macro_component;
  END_ENTITY; -- assembly_module_macro_component_join_terminal

```

```

ENTITY assembly_module_terminal
  SUPERTYPE OF ((ONEOF (assembly_module_interface_terminal,
assembly_module_join_terminal)) ANDOR
(pca_terminal))
  SUBTYPE OF (part_terminal);
    SELF\part_feature.associated_definition : assembly_module_usage_view;
    reference_terminal : OPTIONAL packaged_part_interface_terminal;
    related_connector : OPTIONAL assembly_module_usage_view_connector;
    terminal_connection_zone : OPTIONAL SET[1:?] OF connection_zone;
  INVERSE
    SELF\part_feature.feature_shape : SET[1:?] OF usage_concept_usage_relationship FOR as
  END_ENTITY; -- assembly_module_terminal

```

```

ENTITY assembly_module_usage_view
  SUPERTYPE OF (pca_usage_view)
  SUBTYPE OF (physical_unit_usage_view);
    implemented_function : functional_unit_usage_view;
    maximum_negative_component_height : OPTIONAL datum_based_length_measure;
    maximum_positive_component_height : OPTIONAL datum_based_length_measure;
  INVERSE
    access_mechanisms : SET[2:?] OF assembly_module_terminal FOR associated_definition;
  END_ENTITY; -- assembly_module_usage_view

```

```

ENTITY assembly_module_usage_view_connector_relationship
  SUBTYPE OF (usage_concept_usage_relationship);
    associating_usage : assembly_module_usage_view;
    SELF\usage_concept_usage_relationship.associated_usage : packaged_connector;
    externally_visible_partial_reference_designation : ee_name;
  UNIQUE
    UR1 : externally_visible_partial_reference_designation,
    associating_usage;

```

```
END_ENTITY; -- assembly_module_usage_view_connector_relationship
```

```
ENTITY assembly_requirement_allocation
  SUPERTYPE OF ((thermal_requirement_allocation ANDOR
electromagnetic_compatibility_requirement_allocation) ANDOR
assembly_separation_requirement_allocation)
  SUBTYPE OF (managed_design_object);
  assembly_assignment : component_or_feature;
  requirement         : ee_requirement_occurrence;
END_ENTITY; -- assembly_requirement_allocation
```

```
ENTITY assembly_separation_requirement_allocation
  SUBTYPE OF (assembly_requirement_allocation);
  disjoint_assignment : SET[1:?] OF component_or_feature;
END_ENTITY; -- assembly_separation_requirement_allocation
```

```
ENTITY assembly_technology_specification
  SUBTYPE OF (process_specification);
END_ENTITY; -- assembly_technology_specification
```

```
ENTITY axis_placement
  SUPERTYPE OF (ONEOF (axis_placement_3d,
axis_placement_2d))
  SUBTYPE OF (shape_item);
  reference_direction : direction;
  translation         : cartesian_point;
END_ENTITY; -- axis_placement
```

```
ENTITY axis_placement_2d
  SUBTYPE OF (axis_placement);
END_ENTITY; -- axis_placement_2d
```

```
ENTITY axis_placement_3d
  SUBTYPE OF (axis_placement);
END_ENTITY; -- axis_placement_3d
```

```
ENTITY bare_die
  SUBTYPE OF (physical_unit_usage_view);
  least_material_condition_centroid_location : OPTIONAL cartesian_point;
  maximum_height_above_seating_plane        : OPTIONAL length_data_element;
  maximum_material_condition_centroid_location : OPTIONAL cartesian_point;
  SELF\physical_unit.unit_technology         : technology;
```

```

        implemented_function                : functional_unit_usage_view;
INVERSE
    access_mechanisms : SET[2:?] OF minimally_defined_bare_die_terminal FOR associated_d
    die_seating_plane : non_feature_shape_element FOR scope;
UNIQUE
    UR1 : implemented_function,
unit_technology;
WHERE
    WR1 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' + 'BARE_DIE_TOP_SURFACE' :
    WR2 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' + 'BARE_DIE_BOTTOM_SURFACE' :
    WR3 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PART_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' + 'BARE_DIE_EDGE_SURFACE' :
    WR4 : 'AP210_ARM.SEATING_PLANE' IN TYPEOF(die_seating_plane);
END_ENTITY; -- bare_die

ENTITY bare_die_bottom_surface
    SUBTYPE OF (bare_die_surface);
    SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_bottom_surface

ENTITY bare_die_component
    SUBTYPE OF (physical_component);
    SELF\assembly_component.usage_definition : bare_die;
END_ENTITY; -- bare_die_component

ENTITY bare_die_component_terminal
    SUBTYPE OF (physical_component_terminal);
    SELF\component_feature.definition : bare_die_terminal;
    SELF\component_feature.associated_component : bare_die_component;
END_ENTITY; -- bare_die_component_terminal

ENTITY bare_die_edge_segment_surface
    SUBTYPE OF (bare_die_surface);
    composed_surface : bare_die_edge_surface;
    start_vertex : edge_segment_vertex;
    end_vertex : edge_segment_vertex;
    SELF\part_feature.associated_definition : bare_die;
WHERE
    WR1 : start_vertex :<=: end_vertex;
    WR2 : SELF\part_feature.associated_definition :=: composed_surface\part_feature.asso
END_ENTITY; -- bare_die_edge_segment_surface

ENTITY bare_die_edge_surface

```

```

    SUBTYPE OF (bare_die_surface);
    SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_edge_surface

```

```

ENTITY bare_die_surface
    ABSTRACT SUPERTYPE OF (ONEOF (bare_die_top_surface,
bare_die_bottom_surface,
bare_die_edge_segment_surface,
bare_die_edge_surface))
    SUBTYPE OF (part_mounting_feature);
END_ENTITY; -- bare_die_surface

```

```

ENTITY bare_die_terminal
    SUBTYPE OF (minimally_defined_bare_die_terminal);
    material : ee_material;
    external_connection_area : OPTIONAL SET[1:?] OF connection_zone;
    seating_plane_zone : OPTIONAL connection_zone_interface_plane_relationship;
END_ENTITY; -- bare_die_terminal

```

```

ENTITY bare_die_terminal_surface_constituent_relationship
    SUBTYPE OF (shape_element_constituent_relationship);
    SELF\shape_element_relationship.relate_shape_element : bare_die_surface;
    SELF\shape_element_relationship.related_shape_element : bare_die_terminal;
END_ENTITY; -- bare_die_terminal_surface_constituent_relationship

```

```

ENTITY bare_die_top_surface
    SUBTYPE OF (bare_die_surface);
    SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- bare_die_top_surface

```

```

ENTITY bevel_edge_feature_shape
    SUBTYPE OF (edge_shape);
    minimum_bevel_width : length_data_element;
    minimum_bevel_angle : angle_data_element;
    maximum_bevel_width : length_data_element;
    maximum_bevel_angle : angle_data_element;
END_ENTITY; -- bevel_edge_feature_shape

```

```

ENTITY blind_passage_template
    SUBTYPE OF (unsupported_passage_template);
    of_measure_orientation : measure_orientation;
    datum_reference : stratum_surface;
    bottom_distance_from_datum_reference : shape_dimension_representation;
END_ENTITY; -- blind_passage_template

```



```

ENTITY blind_via
  ABSTRACT SUPERTYPE OF (ONEOF (bonded_conductive_base_blind_via,
non_conductive_base_blind_via,
plated_conductive_base_blind_via))
  SUBTYPE OF (via);
END_ENTITY; -- blind_via

```

```

ENTITY block
  SUBTYPE OF (csg_primitive);
END_ENTITY; -- block

```

```

ENTITY bond_assembly_position
  SUBTYPE OF (managed_design_object);
  bond : assembled_with_bonding;
  assembly_model : physical_unit_3d_shape;
  bond_model : default_assembly_bond_shape;
  bond_definition_placement : axis_placement_3d;
END_ENTITY; -- bond_assembly_position

```

```

ENTITY bonded_conductive_base_blind_via
  SUBTYPE OF (blind_via);
  features_join : stratum_feature_conductive_join;
END_ENTITY; -- bonded_conductive_base_blind_via

```

```

ENTITY boolean_csg_result
  SUBTYPE OF (shape_item);
  operand_1 : boolean_csg_operand;
  operand_2 : boolean_csg_operand;
  csg_operator : boolean_operator;
END_ENTITY; -- boolean_csg_result

```

```

ENTITY bound_volume_shape
  ABSTRACT SUPERTYPE OF (ONEOF (advanced_b_rep,
csg_solid,
edge_based_wireframe,
geometrically_bounded_surface,
shell_based_wireframe,
solid_of_linear_extrusion))
  SUBTYPE OF (shape_definition);
END_ENTITY; -- bound_volume_shape

```

```

ENTITY boundary_based_position_tolerance
  SUBTYPE OF (position_tolerance_with_referenced_datum_system,

```

```

geometric_tolerance_with_applied_material_condition_property);
    profile_control : OPTIONAL profile_of_any_surface_tolerance_with_referenced_datum_sy
END_ENTITY; -- boundary_based_position_tolerance

```

```

ENTITY boundary_size_characteristic
    ABSTRACT SUPERTYPE OF (ONEOF (diametrical_boundary_size_characteristic,
opposing_boundary_set_size_characteristic))
    SUBTYPE OF (managed_design_object);
        tolerance_value      : length_data_element;
        maximum_tolerance_value : OPTIONAL length_data_element;
    INVERSE
        boundary_definition : boundary_zone_definition_with_specified_size FOR specified_bou
END_ENTITY; -- boundary_size_characteristic

```

```

ENTITY boundary_zone_definition_with_specified_size
    SUPERTYPE OF (profile_boundary_definition)
    SUBTYPE OF (tolerance_zone_boundary_definition);
        specified_boundary_size : boundary_size_characteristic;
END_ENTITY; -- boundary_zone_definition_with_specified_size

```

```

ENTITY bounding_curve
    SUBTYPE OF (curve);
        mapping_function : external_definition;
        basis_curve      : trimmed_line;
END_ENTITY; -- bounding_curve

```

```

ENTITY buried_via
    SUBTYPE OF (via);
END_ENTITY; -- buried_via

```

```

ENTITY bus_element_link
    SUBTYPE OF (managed_design_object);
        precedent_element : bus_structural_definition_or_node;
        subsequent_element : bus_structural_definition_or_node;
    UNIQUE
        UR1 : precedent_element,
subsequent_element;
    WHERE
        WR1 : precedent_element <> subsequent_element;
END_ENTITY; -- bus_element_link

```

```

ENTITY bus_structural_definition
    SUBTYPE OF (managed_design_object);
        bus_name      : ee_name;

```

```

        composition : SET[1:?] OF bus_element_link;
    INVERSE
        external_bus_access : SET[0:1] OF functional_unit_network_terminal_definition_bus_as
    UNIQUE
        UR1 : bus_name;
    WHERE
        WR1 : consistent_bus_structural_definition(composition);
END_ENTITY; -- bus_structural_definition

```

```

ENTITY cable_component
    SUBTYPE OF (physical_component);
END_ENTITY; -- cable_component

```

```

ENTITY capacitance_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- capacitance_data_element

```

```

ENTITY cartesian_coordinate_system
    SUBTYPE OF (managed_design_object);
    length_unit          : STRING;
    length_unit_prefix  : OPTIONAL si_prefix;
    length_dynamic_range : OPTIONAL INTEGER;
    dimensionality       : two_or_three_dimensional;
    angle_unit           : degree_or_radian;
    length_uncertainty   : length_data_element;
    angular_uncertainty  : OPTIONAL angle_data_element;
END_ENTITY; -- cartesian_coordinate_system

```

```

ENTITY cartesian_point
    SUBTYPE OF (shape_item);
END_ENTITY; -- cartesian_point

```

```

ENTITY cartesian_transformation_operator_2d
    SUBTYPE OF (managed_design_object);
    axis_1      : direction;
    axis_2      : direction;
    translation : cartesian_point;
    scale       : positive_ratio_measure;
    DERIVE
        mirror : BOOLEAN := derive_mirror(SELF);
END_ENTITY; -- cartesian_transformation_operator_2d

```

```

ENTITY celsius_temperature_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- celsius_temperature_data_element

```

```

ENTITY centre_axis
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_axis

```

```

ENTITY centre_of_symmetry
  ABSTRACT SUPERTYPE OF (ONEOF (centre_axis,
centre_plane,
centre_point))
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- centre_of_symmetry

```

```

ENTITY centre_plane
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_plane

```

```

ENTITY centre_point
  SUBTYPE OF (centre_of_symmetry);
END_ENTITY; -- centre_point

```

```

ENTITY certification
  SUBTYPE OF (managed_design_object);
  certification_approval : ee_approval;
  certification_number   : ee_text;
  certified_product_version : supplied_product_version;
  certification_date      : date_or_date_and_time;
  UNIQUE
  UR1 : certification_number;
END_ENTITY; -- certification

```

```

ENTITY change_design_object_management_relationship
  SUPERTYPE OF (ONEOF (terminal_swap_specification,
gate_path_swap_specification,
component_swap_specification))
  SUBTYPE OF (design_object_management_relationship);
  current_design_object : managed_design_object;
  previous_design_object : managed_design_object;
  UNIQUE
  UR1 : current_design_object,
current_design;
  UR2 : previous_design_object,
previous_design;
END_ENTITY; -- change_design_object_management_relationship

```

```

ENTITY change_order
  SUBTYPE OF (work_order);
    adopted_solution : design_object_management_relationship_or_description;
END_ENTITY; -- change_order

```

```

ENTITY change_request
  SUBTYPE OF (work_request);
    recommended_solution : OPTIONAL SET[1:?] OF design_object_management_relationship_or_description;
    consequence           : SET[1:?] OF ee_text;
    version               : ee_name;
  UNIQUE
    UR1 : version,
  SELF\work_request.work_request_id;
END_ENTITY; -- change_request

```

```

ENTITY characteristic
  SUPERTYPE OF (ONEOF (compound_characteristic,
    numeric_parameter))
  SUBTYPE OF (managed_design_object);
    type_name           : STRING;
    reference_document  : ee_document;
    property_value      : numeric_or_non_numeric_select;
END_ENTITY; -- characteristic

```

```

ENTITY characteristic_data_set
  SUBTYPE OF (compound_characteristic);
    SELF\characteristic.property_value : set_characteristic;
END_ENTITY; -- characteristic_data_set

```

```

ENTITY characteristic_data_table
  SUBTYPE OF (compound_characteristic);
    SELF\characteristic.property_value : list_characteristic;
END_ENTITY; -- characteristic_data_table

```

```

ENTITY circular_area
  SUBTYPE OF (primitive_2d);
    centre : cartesian_point;
    radius : length_data_element;
END_ENTITY; -- circular_area

```

```

ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- circular_runout_tolerance

```

```

ENTITY circularity_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- circularity_tolerance

```

```

ENTITY closed_curve
  SUBTYPE OF (curve);
END_ENTITY; -- closed_curve

```

```

ENTITY common_datum
  SUBTYPE OF (datum);
END_ENTITY; -- common_datum

```

```

ENTITY component_2d_edge_location
  SUBTYPE OF (component_2d_location);
  reference_terminal_assembly_joint : assembly_joint;
  mounting_surface_assembly_joint   : assembly_joint;
  WHERE
    WR1 : reference_terminal_assembly_joint <> mounting_surface_assembly_joint;
    WR2 : SELF\component_2d_location.substrate_location = FALSE;
    WR3 : SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE'] * TYPEOF(mounting_surface_asse
END_ENTITY; -- component_2d_edge_location

```

```

ENTITY component_2d_embedded_location
  SUBTYPE OF (component_2d_location);
  precedent_surface : adjacent_stratum_surface_embedded_component_surface_definition;
  subsequent_surface : adjacent_stratum_surface_embedded_component_surface_definition;
  reference_terminal : fabrication_joint;
  UNIQUE
    UR1 : precedent_surface,
reference_terminal,
subsequent_surface;
  WHERE
    WR1 : precedent_surface <> subsequent_surface;
    WR2 : SELF\component_2d_location.substrate_location = FALSE;
END_ENTITY; -- component_2d_embedded_location

```

```

ENTITY component_2d_location
  SUPERTYPE OF (ONEOF (component_2d_edge_location,
component_2d_embedded_location,
component_2d_stacked_location,
component_2d_surface_location))
  SUBTYPE OF (managed_design_object);
  placement_fixed : BOOLEAN;

```

```

        transformation : cartesian_transformation_operator_2d;
DERIVE
    substrate_location : BOOLEAN := substrate_in_assembly(SELF);
    the_context        : cartesian_coordinate_system := referencing_position.assembly_mo
INVERSE
    referencing_position : component_assembly_2d_position FOR transformation;
WHERE
    WR1 : NOT (substrate_location) OR (transformation.mirror = FALSE);
END_ENTITY; -- component_2d_location

```

```

ENTITY component_2d_stacked_location
    SUBTYPE OF (component_2d_location);
    mounting_joint : assembly_joint;
    WHERE
        WR1 : SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE'] * TYPEOF(mounting_joint.assemb
        WR2 : SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE'] * TYPEOF(mounting_joint.assemb
        WR3 : SELF\component_2d_location.substrate_location = FALSE;
END_ENTITY; -- component_2d_stacked_location

```

```

ENTITY component_2d_surface_location
    SUBTYPE OF (component_2d_location);
    mounting_surface_assembly_joint : assembly_joint;
    WHERE
        WR1 : SELF\component_2d_location.substrate_location = FALSE;
        WR2 : SIZEOF(['AP210_ARM.INTERCONNECT_MODULE_COMPONENT_TERMINAL',
'AP210_ARM.INTERCONNECT_MODULE_COMPONENT_SURFACE_FEATURE'] * TYPEOF(mounting_surface_asse
END_ENTITY; -- component_2d_surface_location

```

```

ENTITY component_3d_location
    SUBTYPE OF (managed_design_object);
    transformation : axis_placement_3d;
    placement_fixed : BOOLEAN;
END_ENTITY; -- component_3d_location

```

```

ENTITY component_assembly_2d_position
    SUBTYPE OF (managed_design_object);
    component_model : assembly_component_2d_shape;
    transformation : component_2d_location;
    assembly_model : physical_unit_planar_shape;
    component      : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_2d_position

```

```

ENTITY component_assembly_3d_position

```

```

SUBTYPE OF (managed_design_object);
  component_model : assembly_component_3d_shape;
  transformation  : component_3d_location;
  assembly_model  : physical_unit_3d_shape;
  component       : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_3d_position

```

```

ENTITY component_assembly_non_planar_position
  SUBTYPE OF (managed_design_object);
    component_model : assembly_component_non_planar_shape;
    transformation  : component_non_planar_location;
    assembly_model  : physical_unit_3d_shape;
    component       : next_higher_assembly_relationship;
END_ENTITY; -- component_assembly_non_planar_position

```

```

ENTITY component_based_annotation_symbol_placement_relationship
  SUBTYPE OF (component_feature_to_layout_feature_relationship);
    SELF\component_feature_to_layout_feature_relationship.related_stratum_concept : spec
END_ENTITY; -- component_based_annotation_symbol_placement_relationship

```

```

ENTITY component_based_annotation_text_relationship
  SUBTYPE OF (component_feature_to_layout_feature_relationship);
    SELF\component_feature_to_layout_feature_relationship.related_stratum_concept : lami
    message_type : comp
END_ENTITY; -- component_based_annotation_text_relationship

```

```

ENTITY component_external_reference
  SUBTYPE OF (managed_design_object);
    design_owner      : STRING;
    part_number       : STRING;
    revision_code     : STRING;
    product_definition_id : STRING;
    reference_designation : STRING;
END_ENTITY; -- component_external_reference

```

```

ENTITY component_feature
  SUPERTYPE OF (ONEOF (physical_component_feature,
    laminate_component_feature))
  SUBTYPE OF (managed_design_object);
    definition : OPTIONAL part_feature_or_template_feature;
    associated_component : assembly_component;
  INVERSE
    defined_product_feature : SET[0:1] OF component_feature_to_physical_usage_view_assign
  UNIQUE
    UR1 : definition,

```



```

associated_component;
END_ENTITY; -- component_feature

```

```

ENTITY component_feature_external_reference
  SUBTYPE OF (managed_design_object);
  associated_component : component_external_reference;
  name                 : STRING;
  UNIQUE
  UR1 : associated_component,
name;
END_ENTITY; -- component_feature_external_reference

```

```

ENTITY component_feature_make_from_relationship
  SUBTYPE OF (managed_design_object);
  reusable_feature : component_feature;
  resultant_feature : component_feature;
END_ENTITY; -- component_feature_make_from_relationship

```

```

ENTITY component_feature_relationship
  SUBTYPE OF (managed_design_object);
  related_component_feature : component_feature;
  relating_component_feature : component_feature;
END_ENTITY; -- component_feature_relationship

```

```

ENTITY component_feature_to_layout_feature_relationship
  SUPERTYPE OF (ONEOF (component_based_annotation_text_relationship,
component_based_annotation_symbol_placement_relationship))
  SUBTYPE OF (managed_design_object);
  related_stratum_concept : stratum_concept;
  relating_feature        : component_or_component_feature;
  requirement             : OPTIONAL ee_requirement_occurrence;
END_ENTITY; -- component_feature_to_layout_feature_relationship

```

```

ENTITY component_feature_to_physical_usage_view_assignment
  SUPERTYPE OF (ONEOF (component_terminal_to_assembly_module_join_terminal_assignment,
component_terminal_to_assembly_module_interface_terminal_assignment,
component_terminal_to_interconnect_module_join_terminal_assignment,
component_terminal_to_interconnect_module_interface_terminal_assignment,
component_terminal_to_pca_terminal_assignment))
  SUBTYPE OF (managed_design_object);
  defined_part_feature : part_feature;
  assigned_design_object : component_feature;
END_ENTITY; -- component_feature_to_physical_usage_view_assignment

```

```

ENTITY component_group_assignment
  SUBTYPE OF (managed_design_object);
    associated_group    : placement_group;
    assigned_component  : assembly_component;
END_ENTITY; -- component_group_assignment

```

```

ENTITY component_interface_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (packaged_connector_component_interface_terminal,
interconnect_component_interface_terminal))
  SUBTYPE OF (physical_component_feature);
    SELF\component_feature.associated_component : interface_component;
END_ENTITY; -- component_interface_terminal

```

```

ENTITY component_make_from_relationship
  SUPERTYPE OF (laminated_group_component_make_from_relationship)
  SUBTYPE OF (managed_design_object);
    reusable_component  : assembly_component;
    resultant_component : assembly_component;
  WHERE
    WR1 : resultant_component <> reusable_component;
END_ENTITY; -- component_make_from_relationship

```

```

ENTITY component_mating_constraint_condition
  SUBTYPE OF (managed_design_object);
    constraining_joint      : SET[1:3] OF assembly_joint;
    constrained_degree_of_freedom : SET[1:5] OF degree_of_freedom;
    of_context              : physical_unit_design_view;
    name                    : ee_name;
  UNIQUE
    UR1 : name,
of_context;
END_ENTITY; -- component_mating_constraint_condition

```

```

ENTITY component_mounting_clearance_relationship
  SUBTYPE OF (managed_design_object);
    feature_relationship      : component_feature_relationship;
    minimum_mounting_clearance : length_data_element;
    maximum_mounting_clearance : OPTIONAL length_data_element;
END_ENTITY; -- component_mounting_clearance_relationship

```

```

ENTITY component_mounting_feature
  SUBTYPE OF (physical_component_feature);
    SELF\component_feature.definition : part_mounting_feature;
END_ENTITY; -- component_mounting_feature

```

```

ENTITY component_non_planar_location
  SUBTYPE OF (managed_design_object);
    transformation : axis_placement_3d;
    placement_fixed : BOOLEAN;
END_ENTITY; -- component_non_planar_location

```

```

ENTITY component_overlap_relationship
  SUBTYPE OF (managed_design_object);
    previously_placed_component : next_higher_assembly_relationship;
    current_component           : next_higher_assembly_relationship;
    minimum_added_clearance     : length_data_element;
    maximum_added_clearance     : OPTIONAL length_data_element;
END_ENTITY; -- component_overlap_relationship

```

```

ENTITY component_part_2d_non_planar_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_planar_shape := SELF\geometric_representation_relationship;
    component_shape : assembly_component_non_planar_shape := SELF\geometric_representation_relationship;
END_ENTITY; -- component_part_2d_non_planar_geometric_representation_relationship

```

```

ENTITY component_part_2d_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_or_physical_unit_2d_shape_select := SELF\geometric_representation_relationship;
    component_shape : assembly_component_2d_shape := SELF\geometric_representation_relationship;
END_ENTITY; -- component_part_2d_geometric_representation_relationship

```

```

ENTITY component_part_3d_geometric_representation_relationship
  SUBTYPE OF (geometric_representation_relationship);
  DERIVE
    part_shape      : part_template_or_physical_unit_3d_shape_select := SELF\geometric_representation_relationship;
    component_shape : assembly_component_3d_shape := SELF\geometric_representation_relationship;
END_ENTITY; -- component_part_3d_geometric_representation_relationship

```

```

ENTITY component_placement_restriction_assignment
  SUBTYPE OF (managed_design_object);
    maximum_negative_component_height : OPTIONAL datum_based_length_measure;
    maximum_positive_component_height : OPTIONAL datum_based_length_measure;
    area                              : OPTIONAL mounting_restriction_area;
    volume                            : OPTIONAL mounting_restriction_volume;
    requirement                       : item_restricted_requirement_occurrence;
    components_permitted              : BOOLEAN;
    maximum_mounting_clearance        : OPTIONAL datum_based_length_measure;

```

```

        minimum_mounting_clearance      : OPTIONAL datum_based_length_measure;
WHERE
    WR1 : EXISTS(area) OR EXISTS(volume);
END_ENTITY; -- component_placement_restriction_assignment

```

```

ENTITY component_swap_specification
    SUBTYPE OF (change_design_object_management_relationship);
    SELF\change_design_object_management_relationship.current_design_object : design_function;
    SELF\change_design_object_management_relationship.previous_design_object : design_function;
END_ENTITY; -- component_swap_specification

```

```

ENTITY component_terminal
    ABSTRACT SUPERTYPE OF (ONEOF (physical_component_terminal,
    laminate_component_join_terminal))
    SUBTYPE OF (component_feature);
    local_swappable : OPTIONAL BOOLEAN;
    global_swappable : OPTIONAL BOOLEAN;
    swap_code : OPTIONAL STRING;
END_ENTITY; -- component_terminal

```

```

ENTITY component_terminal_to_assembly_module_interface_terminal_assignment
    SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
    DERIVE
        usage_view_terminal : assembly_module_interface_terminal := SELF\component_feature_to_physical_usage_view_assignment;
        assigned_terminal : pca_interface_select := SELF\component_feature_to_physical_usage_view_assignment;
    UNIQUE
        UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_assembly_module_interface_terminal_assignment

```

```

ENTITY component_terminal_to_assembly_module_join_terminal_assignment
    SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
    DERIVE
        usage_view_terminal : assembly_module_join_terminal := SELF\component_feature_to_physical_usage_view_assignment;
        assigned_terminal : interconnect_module_macro_component_join_terminal := SELF\component_feature_to_physical_usage_view_assignment;
    UNIQUE
        UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_assembly_module_join_terminal_assignment

```

```

ENTITY component_terminal_to_interconnect_module_interface_terminal_assignment
    SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
    DERIVE
        usage_view_terminal : interconnect_module_interface_terminal := SELF\component_feature_to_physical_usage_view_assignment;
        assigned_terminal : interconnect_or_assembly_module_select := SELF\component_feature_to_physical_usage_view_assignment;
    UNIQUE
        UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_interconnect_module_interface_terminal_assignment

```

```

ENTITY component_terminal_to_interconnect_module_join_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
  DERIVE
    usage_view_terminal : interconnect_module_join_terminal := SELF\component_feature_to_
    assigned_terminal   : fabricated_feature_select := SELF\component_feature_to_physical_
  UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_interconnect_module_join_terminal_assignment

```

```

ENTITY component_terminal_to_pca_terminal_assignment
  SUBTYPE OF (component_feature_to_physical_usage_view_assignment);
  DERIVE
    usage_view_terminal : pca_terminal := SELF\component_feature_to_physical_usage_view_
    assigned_terminal   : pca_interface_select := SELF\component_feature_to_physical_usa
  UNIQUE
    UR1 : usage_view_terminal;
END_ENTITY; -- component_terminal_to_pca_terminal_assignment

```

```

ENTITY component_termination_passage
  SUBTYPE OF (plated_passage);
  SELF\assembly_component.usage_definition : component_termination_passage_template;
  INVERSE
    access_mechanism          : SET[1:?] OF component_termination_passage_join_terminal
    interface_access_mechanism : component_termination_passage_interface_terminal FOR as
END_ENTITY; -- component_termination_passage

```

```

ENTITY component_termination_passage_and_contact_size_dependent_land
  SUBTYPE OF (land);
  SELF\assembly_component.usage_definition : default_attachment_size_and_component_term
  reference_passage                        : component_termination_passage;
  INVERSE
    interface_access_mechanism : land_interface_terminal FOR associated_component;
END_ENTITY; -- component_termination_passage_and_contact_size_dependent_land

```

```

ENTITY component_termination_passage_and_contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  SELF\assembly_component.usage_definition : default_attachment_size_and_component_term
  reference_passage                        : component_termination_passage;
  INVERSE
    interface_access_mechanism : non_functional_land_interface_terminal FOR associated_c
END_ENTITY; -- component_termination_passage_and_contact_size_dependent_non_functional_l

```

```

ENTITY component_termination_passage_dependent_land
  SUBTYPE OF (land);

```

```

        reference_passage                : component_termination_passage;
        SELF\assembly_component.usage_definition : default_component_termination_passage_base;
END_ENTITY; -- component_termination_passage_dependent_land

```

```

ENTITY component_termination_passage_dependent_non_functional_land
    SUBTYPE OF (non_functional_land);
        reference_passage                : component_termination_passage;
        SELF\assembly_component.usage_definition : default_component_termination_passage_base;
END_ENTITY; -- component_termination_passage_dependent_non_functional_land

```

```

ENTITY component_termination_passage_interface_terminal
    SUBTYPE OF (laminar_component_interface_terminal);
        SELF\component_feature.definition      : component_termination_passage_template;
        SELF\component_feature.associated_component : component_termination_passage;
END_ENTITY; -- component_termination_passage_interface_terminal

```

```

ENTITY component_termination_passage_join_terminal
    SUBTYPE OF (laminar_component_join_terminal);
        SELF\component_feature.definition      : component_termination_passage_template;
        SELF\component_feature.associated_component : component_termination_passage;
END_ENTITY; -- component_termination_passage_join_terminal

```

```

ENTITY component_termination_passage_template
    SUBTYPE OF (inter_stratum_feature_template);
        SELF\inter_stratum_feature_template.of_passage_technology : default_component_termination_passage_template;
    INVERSE
        access_mechanisms : SET[2:?] OF component_termination_passage_template_terminal FOR a;
END_ENTITY; -- component_termination_passage_template

```

```

ENTITY component_termination_passage_template_interface_terminal
    SUBTYPE OF (component_termination_passage_template_terminal);
END_ENTITY; -- component_termination_passage_template_interface_terminal

```

```

ENTITY component_termination_passage_template_join_terminal
    SUBTYPE OF (component_termination_passage_template_terminal);
        disallowed_inter_stratum_extent : OPTIONAL SET[1:?] OF inter_stratum_extent;
END_ENTITY; -- component_termination_passage_template_join_terminal

```

```

ENTITY component_termination_passage_template_terminal
    ABSTRACT SUPERTYPE OF (ONEOF (component_termination_passage_template_interface_terminal,
    component_termination_passage_template_join_terminal))
    SUBTYPE OF (shape_element);

```

```

        associated_definition      : component_termination_passage_template;
        terminal_connection_zone   : OPTIONAL SET[1:?] OF connection_zone;
        SELF\shape_element.name    : ee_name;
    INVERSE
        terminal_shape : SET[0:1] OF usage_concept_usage_relationship FOR associated_usage;
    UNIQUE
        UR1 : name,
associated_definition;
END_ENTITY; -- component_termination_passage_template_terminal

```

```

ENTITY compound_characteristic
    ABSTRACT SUPERTYPE OF (ONEOF (characteristic_data_set,
characteristic_data_table))
    SUBTYPE OF (characteristic);
        SELF\characteristic.property_value : compound_characteristic_definition;
END_ENTITY; -- compound_characteristic

```

```

ENTITY composite_curve
    SUBTYPE OF (curve);
        curve_segment : LIST[1:?] OF UNIQUE composite_curve_segment;
END_ENTITY; -- composite_curve

```

```

ENTITY composite_curve_segment
    SUBTYPE OF (managed_design_object);
        segment_curve : curve;
    INVERSE
        using_curves : SET[1:?] OF composite_curve FOR curve_segment;
END_ENTITY; -- composite_curve_segment

```

```

ENTITY composite_feature_locating_position_tolerance
    SUBTYPE OF (position_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_feature_locating_position_tolerance

```

```

ENTITY composite_feature_relating_position_tolerance
    SUBTYPE OF (position_tolerance);
        refined_control : locating_or_relating_position_tolerance;
END_ENTITY; -- composite_feature_relating_position_tolerance

```

```

ENTITY composite_group_shape_element
    SUBTYPE OF (composite_shape_element);
END_ENTITY; -- composite_group_shape_element

```

```

ENTITY composite_linear_profile_locating_tolerance
  SUBTYPE OF (profile_of_any_line_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_linear_profile_locating_tolerance

```

```

ENTITY composite_linear_profile_refinement_tolerance
  SUBTYPE OF (profile_of_any_line_tolerance);
  refined_control : locating_or_refinement_linear_profile_tolerance;
END_ENTITY; -- composite_linear_profile_refinement_tolerance

```

```

ENTITY composite_shape_element
  ABSTRACT SUPERTYPE OF (ONEOF (composite_group_shape_element,
composite_unit_shape_element))
  SUBTYPE OF (shape_element);
  INVERSE
    composing_relationships : SET[2:?] OF shape_element_composing_relationship FOR relat
END_ENTITY; -- composite_shape_element

```

```

ENTITY composite_signal_property
  SUBTYPE OF (signal_property);
  INVERSE
    signal_property_structure : SET[1:?] OF composite_signal_property_relationship FOR c
END_ENTITY; -- composite_signal_property

```

```

ENTITY composite_signal_property_relationship
  SUBTYPE OF (managed_design_object);
  composition : composite_signal_property;
  signal_property_structural_element : signal_property;
  aggregation_operation : external_definition;
END_ENTITY; -- composite_signal_property_relationship

```

```

ENTITY composite_surface_profile_locating_tolerance
  SUBTYPE OF (profile_of_any_surface_tolerance_with_referenced_datum_system);
END_ENTITY; -- composite_surface_profile_locating_tolerance

```

```

ENTITY composite_surface_profile_refinement_tolerance
  SUBTYPE OF (profile_of_any_surface_tolerance);
  refined_control : locating_or_refinement_surface_profile_tolerance;
END_ENTITY; -- composite_surface_profile_refinement_tolerance

```

```

ENTITY composite_unit_shape_element
  SUBTYPE OF (composite_shape_element);
END_ENTITY; -- composite_unit_shape_element

```



```

ENTITY concentricity_and_coaxiality_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- concentricity_and_coaxiality_tolerance

```

```

ENTITY conductance_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- conductance_data_element

```

```

ENTITY conductive_filled_area
  SUPERTYPE OF (connected_filled_area)
  SUBTYPE OF (stratum_feature);
END_ENTITY; -- conductive_filled_area

```

```

ENTITY conductive_interconnect_element
  ABSTRACT SUPERTYPE OF (ONEOF (unrouted_conductive_interconnect_element,
conductive_interconnect_element_with_pre_defined_transitions,
conductive_interconnect_element_with_user_defined_single_transition))
  SUBTYPE OF (laminate_component);
  composed_conductor : OPTIONAL stratum_feature;
  INVERSE
    access_mechanisms : SET[2:?] OF conductive_interconnect_element_terminal FOR association;
END_ENTITY; -- conductive_interconnect_element

```

```

ENTITY conductive_interconnect_element_terminal
  SUBTYPE OF (laminate_component_feature);
  SELF\component_feature.associated_component : conductive_interconnect_element;
  connection : OPTIONAL connection_zone;
  name : STRING;
  location : layer_connection_point;
  UNIQUE
    UR1 : name,
SELF\component_feature.associated_component;
  WHERE
    WR1 : NOT (EXISTS(SELF\component_feature.associated_component\conductive_interconnect_element_terminal));
END_ENTITY; -- conductive_interconnect_element_terminal

```

```

ENTITY conductive_interconnect_element_terminal_link
  SUBTYPE OF (managed_design_object);
  associated_element : conductive_interconnect_element_with_pre_defined_transitions;
  precedent_point : conductive_interconnect_element_terminal;
  subsequent_point : conductive_interconnect_element_terminal;
  WHERE
    WR1 : precedent_point <> subsequent_point;
    WR2 : precedent_point.associated_component = associated_element;

```

```

        WR3 : subsequent_point.associated_component = associated_element;
    END_ENTITY; -- conductive_interconnect_element_terminal_link

```

```

ENTITY conductive_interconnect_element_with_pre_defined_transitions
    SUPERTYPE OF (join_two_physical_connectivity_definition_supporting_printed_component)
    SUBTYPE OF (conductive_interconnect_element);
        SELF\assembly_component.usage_definition          : trace_template;
        centreline_shape                                  : curve;
        SELF\conductive_interconnect_element.composed_conductor : stratum_feature;
    INVERSE
        connected_points : SET[1:?] OF conductive_interconnect_element_terminal_link FOR ass
    WHERE
        WR1 : SIZEOF(connected_points) = SIZEOF(SELF\conductive_interconnect_element.access_
        WR2 : correlated_terminals(SELF\conductive_interconnect_element.access_mechanisms,
connected_points);
    END_ENTITY; -- conductive_interconnect_element_with_pre_defined_transitions

```

```

ENTITY conductive_interconnect_element_with_user_defined_single_transition
    SUBTYPE OF (conductive_interconnect_element);
        SELF\conductive_interconnect_element.composed_conductor : stratum_feature;
        start_definition                                          : trace_template;
        end_definition                                             : trace_template;
        left_curve_shape_representation                           : trimmed_or_bounding_curve;
        right_curve_shape_representation                          : trimmed_or_bounding_curve;
        start_terminus                                             : layer_connection_point;
        end_terminus                                               : layer_connection_point;
    END_ENTITY; -- conductive_interconnect_element_with_user_defined_single_transition

```

```

ENTITY conductor
    SUBTYPE OF (stratum_feature);
        SELF\stratum_feature.resident_stratum : design_layer_stratum;
        implementation_requirement             : intra_stratum_join_relationship;
    END_ENTITY; -- conductor

```

```

ENTITY configuration_controlled_message
    SUBTYPE OF (managed_design_object);
        message_source          : message_source_select;
        message_content          : STRING;
        message_type_specification : message_type;
    END_ENTITY; -- configuration_controlled_message

```

```

ENTITY conical_tolerance_zone_boundary
    SUBTYPE OF (tolerance_zone_boundary);
    INVERSE
        boundary_and_surface_relationships : SET[2:2] OF conical_tolerance_zone_boundary_and_
    END_ENTITY; -- conical_tolerance_zone_boundary

```

```

ENTITY conical_tolerance_zone_boundary_and_surface_relationship
  SUBTYPE OF (managed_design_object);
    sized_tolerance_zone_boundary : conical_tolerance_zone_boundary;
    terminating_surface           : physical_feature_or_part_template;
  INVERSE
    size_characteristic : conical_tolerance_zone_boundary_diametrical_size_characteristic;
END_ENTITY; -- conical_tolerance_zone_boundary_and_surface_relationship

ENTITY conical_tolerance_zone_boundary_definition
  SUBTYPE OF (tolerance_zone_boundary_definition);
    boundary_sizes : SET[2:2] OF conical_tolerance_zone_boundary_definition;
    SELF\tolerance_zone_boundary_definition.defined_zone_boundary : conical_tolerance_zone_boundary_definition;
END_ENTITY; -- conical_tolerance_zone_boundary_definition

ENTITY conical_tolerance_zone_boundary_diametrical_size_characteristic
  SUBTYPE OF (managed_design_object);
    sized_boundary_at_surface : conical_tolerance_zone_boundary_and_surface_relationship;
    tolerance_value           : length_data_element;
  INVERSE
    boundary_definition : conical_tolerance_zone_boundary_definition FOR boundary_sizes;
END_ENTITY; -- conical_tolerance_zone_boundary_diametrical_size_characteristic

ENTITY conical_tolerance_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
    SELF\tolerance_zone_definition.boundary_definition : conical_tolerance_zone_boundary_definition;
END_ENTITY; -- conical_tolerance_zone_definition

ENTITY connected_area_component
  SUBTYPE OF (stratum_feature_template_component);
    associated_layer_connection_point : SET[1:?] OF layer_connection_point;
  WHERE
    WR1 : NOT (EXISTS(SELF\assembly_component.usage_definition));
END_ENTITY; -- connected_area_component

ENTITY connected_filled_area
  SUBTYPE OF (conductive_filled_area);
    SELF\stratum_feature.resident_stratum : design_layer_stratum;
    implementation_requirement            : intra_stratum_join_relationship;
END_ENTITY; -- connected_filled_area

ENTITY connection_zone
  SUBTYPE OF (managed_design_object);
    representation_colour : OPTIONAL ee_colour;

```

```

INVERSE
    zone_shape_1 : SET[0:1] OF design_view_connection_zone_position_relationship FOR ass
    zone_shape_2 : SET[0:1] OF usage_view_connection_zone_terminal_shape_relationship FOR
WHERE
    WR1 : SIZEOF(USEDIN(SELF,
'AP210_ARM.ASSEMBLY_MODULE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.BARE_DIE_TERMINAL.EXTERNAL_CONNECTION_AREA')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + S
'AP210_ARM.CONDUCTIVE_INTERCONNECT_ELEMENT_TERMINAL.CONNECTION')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.INTERCONNECT_COMPONENT_INTERFACE_TERMINAL.CONNECTION')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.INTERCONNECT_COMPONENT_JOIN_TERMINAL.CONNECTION')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.INTERCONNECT_MODULE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.LAND_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.PACKAGE_TERMINAL.EXTERNAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.PACKAGE_TERMINAL.INTERNAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.PRINTED_PART_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SEL
'AP210_ARM.VIA_TEMPLATE_TERMINAL.TERMINAL_CONNECTION_ZONE')) + SIZEOF(USEDIN(SELF,
'AP210_ARM.PART_MOUNTING_FEATURE.MOUNTING_AREA')) >= 0;
    WR2 : SIZEOF(zone_shape_1) + SIZEOF(zone_shape_2) = 1;
END_ENTITY; -- connection_zone

```

```

ENTITY connection_zone_based_assembly_joint
    SUBTYPE OF (assembly_joint);
    zone_1 : connection_zone;
    zone_2 : connection_zone;
WHERE
    WR1 : zone_1 :<>: zone_2;
END_ENTITY; -- connection_zone_based_assembly_joint

```

```

ENTITY connection_zone_based_fabrication_joint
    SUBTYPE OF (fabrication_joint);
    zone_1 : connection_zone;
    zone_2 : connection_zone;
WHERE
    WR1 : zone_1 :<>: zone_2;
END_ENTITY; -- connection_zone_based_fabrication_joint

```

```

ENTITY connection_zone_interface_plane_relationship
    SUBTYPE OF (managed_design_object);
    interface_plane      : seating_plane;
    mating_zone          : connection_zone;
    derived_zone         : connection_zone;
    derived_zone_shape_class : STRING;
INVERSE
    associated_package_terminal : SET[0:1] OF package_terminal FOR seating_plane_zone;
    associated_bare_die_terminal : SET[0:1] OF bare_die_terminal FOR seating_plane_zone;
UNIQUE
    UR1 : interface_plane,

```

```

mating_zone;
  WHERE
    WR1 : derived_zone_shape_class IN ['area',
'edge'];
    WR2 : NOT ((SIZEOF(associated_package_terminal) = 1) AND (SIZEOF(associated_bare_die
END_ENTITY; -- connection_zone_interface_plane_relationship

ENTITY connection_zone_map_identification
  SUBTYPE OF (managed_design_object);
    reference_connection_zone_shape : usage_view_connection_zone_terminal_shape_relation;
    mapped_connection_zone_shape    : usage_view_connection_zone_terminal_shape_relation;
  UNIQUE
    UR1 : reference_connection_zone_shape,
mapped_connection_zone_shape;
  WHERE
    WR1 : reference_connection_zone_shape <> mapped_connection_zone_shape;
    WR2 : reference_connection_zone_shape.associated_connection_zone_shape_definition <>
END_ENTITY; -- connection_zone_map_identification

ENTITY connection_zone_model_port_assignment
  SUBTYPE OF (managed_design_object);
    source : connection_zone;
    target : analytical_model_port;
END_ENTITY; -- connection_zone_model_port_assignment

ENTITY connectivity_allocation
  SUBTYPE OF (managed_design_object);
    implementation      : physical_connectivity_definition;
    connectivity_requirement : aggregate_connectivity_requirement;
  UNIQUE
    UR1 : implementation;
END_ENTITY; -- connectivity_allocation

ENTITY connectivity_allocation_to_physical_network
  SUBTYPE OF (managed_design_object);
    implementation      : physical_network;
    connectivity_requirement : aggregate_connectivity_requirement;
  UNIQUE
    UR1 : implementation;
END_ENTITY; -- connectivity_allocation_to_physical_network

ENTITY constraint_occurrence
  ABSTRACT SUPERTYPE OF (ee_product_definition_based_constraint_occurrence ANDOR
rule_based_constraint_occurrence)
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- constraint_occurrence

```

```

ENTITY contact_size_dependent_land
  SUBTYPE OF (land);
  SELF\assembly_component.usage_definition : default_attachment_size_based_land_physics;
END_ENTITY; -- contact_size_dependent_land

```

```

ENTITY contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  SELF\assembly_component.usage_definition : default_attachment_size_based_land_physics;
END_ENTITY; -- contact_size_dependent_non_functional_land

```

```

ENTITY contract
  SUBTYPE OF (managed_design_object);
  contractor      : OPTIONAL person_organization;
  contract_number : ee_text;
  contract_approval : OPTIONAL ee_approval;
  contract_date   : OPTIONAL date_or_date_and_time;
  UNIQUE
    UR1 : contract_number;
END_ENTITY; -- contract

```

```

ENTITY coordinated_characteristic
  SUPERTYPE OF (ONEOF (range_characteristic,
tolerance_characteristic))
  SUBTYPE OF (managed_design_object);
  type_name      : STRING;
  component_characteristic : SET[1:?] OF characteristic;
  dependent_environment : OPTIONAL STRING;
  reference_document : ee_document;
END_ENTITY; -- coordinated_characteristic

```

```

ENTITY counterbore_passage_template
  SUBTYPE OF (unsupported_passage_template);
  smaller_passage : unsupported_passage_template;
  larger_passage  : blind_passage_template;
END_ENTITY; -- counterbore_passage_template

```

```

ENTITY countersunk_passage_template
  SUBTYPE OF (unsupported_passage_template);
  constant_diameter_passage : unsupported_passage_template;
  tapered_passage           : blind_passage_template;
END_ENTITY; -- countersunk_passage_template

```

```

ENTITY csg_2d_shape

```

```

    SUBTYPE OF (planar_shape);
    WHERE
        WR1 : SIZEOF(QUERY(c <* SELF\shape_definition.element | 'AP210_ARM.CURVE' IN TYPEOF(
END_ENTITY; -- csg_2d_shape

```

```

ENTITY csg_primitive
    SUPERTYPE OF (ONEOF (block,
right_circular_cylinder,
right_angular_wedge))
    SUBTYPE OF (managed_design_object);
END_ENTITY; -- csg_primitive

```

```

ENTITY csg_solid
    SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- csg_solid

```

```

ENTITY curve
    SUPERTYPE OF ((ONEOF (closed_curve,
bounding_curve) ANDOR
trimmed_line) ANDOR
composite_curve)
    SUBTYPE OF (shape_item);
END_ENTITY; -- curve

```

```

ENTITY curve_dimension
    SUBTYPE OF (single_boundary_dimension);
END_ENTITY; -- curve_dimension

```

```

ENTITY curve_style
    SUBTYPE OF (managed_design_object);
    name : STRING;
    corner_style : extend_or_chord_2_extend_or_truncate_or_round;
    curve_width : length_data_element;
    curve_font : OPTIONAL curve_font_or_scaled_curve_font_select;
    end_extension : extend_or_truncate;
    end_style : square_or_round;
    width_uncertainty : length_data_element;
    UNIQUE
        UR1 : name;
END_ENTITY; -- curve_style

```

```

ENTITY curve_style_font
    SUBTYPE OF (managed_design_object);
END_ENTITY; -- curve_style_font

```

```

ENTITY curve_style_font_and_scaling
  SUBTYPE OF (managed_design_object);
    curve_font      : curve_style_font_select;
    name            : STRING;
    curve_font_scaling : positive_ratio_measure;
  UNIQUE
    UR1 : name;
END_ENTITY; -- curve_style_font_and_scaling

```

```

ENTITY cutout
  SUPERTYPE OF (ONEOF (physical_connectivity_interrupting_cutout,
    plated_cutout,
    partially_plated_cutout))
  SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- cutout

```

```

ENTITY cutout_edge_segment
  SUPERTYPE OF (plated_cutout_edge_segment)
  SUBTYPE OF (inter_stratum_feature);
    composed_cutout      : cutout;
    start_vertex          : edge_segment_vertex;
    end_vertex            : edge_segment_vertex;
    SELF\inter_stratum_feature.usage_definition : inter_stratum_feature_edge_segment_tem
  WHERE
    WR1 : start_vertex :<>: end_vertex;
END_ENTITY; -- cutout_edge_segment

```

```

ENTITY cylindricity_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- cylindricity_tolerance

```

```

ENTITY data_element
  SUPERTYPE OF (ONEOF (absorbed_dose_data_element,
    activity_data_element,
    angle_data_element,
    area_data_element,
    capacitance_data_element,
    celsius_temperature_data_element,
    conductance_data_element,
    dose_equivalent_data_element,
    electric_charge_data_element,
    electric_current_data_element,
    electromotive_force_data_element,
    energy_data_element,
    event_data_element,
    force_data_element,

```



```

frequency_data_element,
illuminance_data_element,
inductance_data_element,
length_data_element,
luminous_flux_data_element,
luminous_intensity_data_element,
magnetic_flux_data_element,
magnetic_flux_density_data_element,
mass_data_element,
power_data_element,
pressure_data_element,
ratio_data_element,
resistance_data_element,
scattering_parameter_data_element,
solid_angle_data_element,
thermodynamic_temperature_data_element,
time_data_element,
volume_data_element))
    SUBTYPE OF (managed_design_object);
        measure_value          : REAL;
        unit_of_measure         : STRING;
        unit_of_measure_prefix : OPTIONAL si_prefix;
    END_ENTITY; -- data_element

```

```

ENTITY date
    SUPERTYPE OF (date_and_time)
    SUBTYPE OF (managed_design_object);
        day    : INTEGER;
        year   : INTEGER;
        month   : month_representation;
    END_ENTITY; -- date

```

```

ENTITY date_and_time
    SUBTYPE OF (date);
        hour       : INTEGER;
        minute     : INTEGER;
        second     : INTEGER;
        time_zone  : INTEGER;
        time_zone_sense : ahead_or_behind;
    END_ENTITY; -- date_and_time

```

```

ENTITY datum
    ABSTRACT SUPERTYPE OF (ONEOF (datum_point,
datum_axis,
datum_plane) ANDOR
ONEOF (single_datum,
common_datum))
    SUBTYPE OF (non_feature_shape_element);

```

```

    INVERSE
    datum_feature_usages : SET[0:?] OF datum_feature_usage_in_datum_relationship FOR rel
END_ENTITY; -- datum

```

```

ENTITY datum_axis
    SUBTYPE OF (datum);
END_ENTITY; -- datum_axis

```

```

ENTITY datum_axis_and_tolerance_zone_orienting_relationship
    SUBTYPE OF (managed_design_object);
    oriented_tolerance_zone : tolerance_zone;
    orienting_datum_axis    : datum_axis;
    INVERSE
    zone_orientation : datum_axis_related_orientation FOR orienting_relationship;
END_ENTITY; -- datum_axis_and_tolerance_zone_orienting_relationship

```

```

ENTITY datum_axis_related_orientation
    SUBTYPE OF (managed_design_object);
    orienting_relationship : datum_axis_and_tolerance_zone_orienting_relationship;
    orientation_type       : centre_type;
END_ENTITY; -- datum_axis_related_orientation

```

```

ENTITY datum_based_direction_vector
    SUBTYPE OF (managed_design_object);
    direction_vector_representation : axis_placement;
    vector_of                       : datum_plane;
    positive_direction_vector       : BOOLEAN;
END_ENTITY; -- datum_based_direction_vector

```

```

ENTITY datum_based_length_measure
    SUBTYPE OF (managed_design_object);
    measure : length_data_element;
    measure_orientation : descriptive_or_geometric_orientation;
    of_datum : datum_plane;
END_ENTITY; -- datum_based_length_measure

```

```

ENTITY datum_based_vector_orientation
    SUBTYPE OF (managed_design_object);
    direction_components : SET[2:2] OF direction_element;
END_ENTITY; -- datum_based_vector_orientation

```

```

ENTITY datum_difference

```

```

SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.name : label;
DERIVE
  precedent_datum : datum := SELF\shape_element_relationship.relate_shape_element;
  subsequent_datum : datum := SELF\shape_element_relationship.related_shape_element;
END_ENTITY; -- datum_difference

ENTITY datum_difference_analytical_model_port_assignment
  SUBTYPE OF (managed_design_object);
    assigned_analytical_model_port : analytical_model_port;
    of_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_analytical_model_port_assignment

ENTITY datum_difference_based_characteristic
  SUBTYPE OF (characteristic);
    of_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_based_characteristic

ENTITY datum_difference_based_model_parameter
  SUBTYPE OF (model_parameter);
    of_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_based_model_parameter

ENTITY datum_difference_functional_unit_usage_view_terminal_assignment
  SUBTYPE OF (managed_design_object);
    assigned_functional_terminal_definition : functional_unit_usage_view_terminal_definition;
    assigned_datum_difference : datum_difference;
END_ENTITY; -- datum_difference_functional_unit_usage_view_terminal_assignment

ENTITY datum_feature
  SUPERTYPE OF (datum_target_set)
  SUBTYPE OF (physical_feature_or_part_template);
    identification : label;
  INVERSE
    datum_feature_usages : SET[1:?] OF datum_feature_usage_in_datum_relationship FOR relationship;
END_ENTITY; -- datum_feature

ENTITY datum_feature_material_condition_property
  SUBTYPE OF (managed_design_object);
    applied_to : datum_feature_usage_in_datum_system_relationship;
    name : material_condition_property_names;
  INVERSE
    datum_system_conditions_definition : datum_system_defining_conditions_with_material_conditions;
END_ENTITY; -- datum_feature_material_condition_property

```

```

ENTITY datum_feature_usage_in_common_datum_relationship
  SUBTYPE OF (datum_feature_usage_in_datum_relationship);
  SELF\shape_element_relationship.relatng_shape_element : common_datum;
END_ENTITY; -- datum_feature_usage_in_common_datum_relationship

```

```

ENTITY datum_feature_usage_in_datum_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (datum_feature_usage_in_single_datum_relationship,
datum_feature_usage_in_common_datum_relationship))
  SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.relatng_shape_element : datum;
  SELF\shape_element_relationship.related_shape_element : datum_feature;
  UNIQUE
    UR1 : related_shape_element,
relatng_shape_element;
END_ENTITY; -- datum_feature_usage_in_datum_relationship

```

```

ENTITY datum_feature_usage_in_datum_system_relationship
  SUBTYPE OF (managed_design_object);
  established_datum_system : datum_system;
  used_datum_feature      : datum_feature;
  INVERSE
    applied_material_condition_property : datum_feature_material_condition_property FOR a
END_ENTITY; -- datum_feature_usage_in_datum_system_relationship

```

```

ENTITY datum_feature_usage_in_single_datum_relationship
  SUBTYPE OF (datum_feature_usage_in_datum_relationship);
  SELF\shape_element_relationship.relatng_shape_element : single_datum;
END_ENTITY; -- datum_feature_usage_in_single_datum_relationship

```

```

ENTITY datum_plane
  SUBTYPE OF (datum);
END_ENTITY; -- datum_plane

```

```

ENTITY datum_point
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (datum);
  location_in_2d : OPTIONAL cartesian_point;
  location_in_3d : OPTIONAL cartesian_point;
  associated_volume_shape_intersection : OPTIONAL volume_shape_intersection;
  WHERE
    WR1 : NOT (EXISTS(location_in_2d) AND EXISTS(location_in_3d)) OR EXISTS(associated_v
    WR2 : NOT (EXISTS(location_in_2d)) OR (location_in_2d.geometric_context.dimensionali
    WR3 : NOT (EXISTS(location_in_3d)) OR (location_in_3d.geometric_context.dimensionali
END_ENTITY; -- datum_point

```

```

ENTITY datum_precedence_assignment
  SUBTYPE OF (managed_design_object);
    assigned_to : datum_usage_in_datum_system_relationship;
    name       : precedence_type;
  INVERSE
    datum_system_conditions_definition : datum_system_defining_conditions FOR assigned_d
  UNIQUE
    UR1 : datum_system_conditions_definition,
name;
END_ENTITY; -- datum_precedence_assignment

```

```

ENTITY datum_reference_frame
  SUBTYPE OF (non_feature_shape_element);
    reference_planes      : SET[3:3] OF datum_plane;
    reference_plane_pairs : SET[3:3] OF perpendicular_shape_element_relationship;
    reference_axes        : SET[3:3] OF datum_axis;
    reference_axis_pairs  : SET[3:3] OF perpendicular_shape_element_relationship;
    reference_origin      : datum_point;
  INVERSE
    used_datum_systems : SET[1:?] OF datum_system FOR established_datum_reference_frame;
  WHERE
    WR1 : 'AP210_ARM.DERIVED_SHAPE_ELEMENT' IN TYPEOF(reference_origin);
    WR6 : SIZEOF(QUERY(ra <* reference_axes | NOT ('AP210_ARM.DERIVED_SHAPE_ELEMENT' IN
END_ENTITY; -- datum_reference_frame

```

```

ENTITY datum_system
  SUBTYPE OF (non_feature_shape_element);
    established_datum_reference_frame : datum_reference_frame;
  INVERSE
    datum_usages      : SET[1:3] OF datum_usage_in_datum_system_relationship FOR estab
    defining_conditions : datum_system_defining_conditions FOR conditions_of;
END_ENTITY; -- datum_system

```

```

ENTITY datum_system_defining_conditions
  ABSTRACT SUPERTYPE OF (ONEOF (datum_system_defining_conditions_with_material_conditions,
datum_system_defining_conditions_without_material_conditions))
  SUBTYPE OF (managed_design_object);
    conditions_of      : datum_system;
    assigned_datum_precedences : SET[1:3] OF datum_precedence_assignment;
END_ENTITY; -- datum_system_defining_conditions

```

```

ENTITY datum_system_defining_conditions_with_material_conditions
  SUBTYPE OF (datum_system_defining_conditions);
    applied_material_condition_properties : SET[1:?] OF datum_feature_material_condition
END_ENTITY; -- datum_system_defining_conditions_with_material_conditions

```

```

ENTITY datum_system_defining_conditions_without_material_conditions
  SUBTYPE OF (datum_system_defining_conditions);
END_ENTITY; -- datum_system_defining_conditions_without_material_conditions

```

```

ENTITY datum_target
  SUBTYPE OF (physical_feature_or_part_template);
  INVERSE
    datum_target_usages : SET[1:?] OF datum_target_usage_in_datum_target_set_relationship;
END_ENTITY; -- datum_target

```

```

ENTITY datum_target_set
  SUBTYPE OF (datum_feature);
  INVERSE
    included_datum_targets : SET[1:?] OF datum_target_usage_in_datum_target_set_relationship;
END_ENTITY; -- datum_target_set

```

```

ENTITY datum_target_usage_in_datum_target_set_relationship
  SUBTYPE OF (managed_design_object);
  defined_datum_target_set : datum_target_set;
  used_datum_target       : datum_target;
  datum_target_number     : positive_integer;
  UNIQUE
    UR1 : used_datum_target,
defined_datum_target_set;
    UR2 : datum_target_number,
defined_datum_target_set;
END_ENTITY; -- datum_target_usage_in_datum_target_set_relationship

```

```

ENTITY datum_usage_in_datum_system_relationship
  SUBTYPE OF (managed_design_object);
  established_datum_system : datum_system;
  used_datum              : datum;
  INVERSE
    precedence_assignment : datum_precedence_assignment FOR assigned_to;
  UNIQUE
    UR1 : used_datum,
established_datum_system;
END_ENTITY; -- datum_usage_in_datum_system_relationship

```

```

ENTITY default_assembly_bond_shape
  SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
  characterized_bond_definition : assembly_bond_definition;
END_ENTITY; -- default_assembly_bond_shape

```

```

ENTITY default_attachment_size_and_component_termination_passage_based_land_physical_tem
    SUBTYPE OF (default_component_termination_passage_based_land_physical_template,
default_attachment_size_based_land_physical_template);
END_ENTITY; -- default_attachment_size_and_component_termination_passage_based_land_phys.

```

```

ENTITY default_attachment_size_and_via_based_land_physical_template
    SUBTYPE OF (default_via_based_land_physical_template,
default_attachment_size_based_land_physical_template);
END_ENTITY; -- default_attachment_size_and_via_based_land_physical_template

```

```

ENTITY default_attachment_size_based_land_physical_template
    SUPERTYPE OF (ONEOF (default_attachment_size_and_component_termination_passage_based_l
default_attachment_size_and_via_based_land_physical_template))
    SUBTYPE OF (default_land_physical_template);
        minimum_attachment_region_size : OPTIONAL connection_zone;
        maximum_attachment_region_size : OPTIONAL connection_zone;
    INVERSE
        interface_access_mechanisms : SET[1:?] OF land_template_interface_terminal FOR assoc
    WHERE
        WR1 : NOT (EXISTS(minimum_attachment_region_size) AND EXISTS(maximum_attachment_regi
END_ENTITY; -- default_attachment_size_based_land_physical_template

```

```

ENTITY default_component_termination_passage_based_land_physical_template
    SUPERTYPE OF (default_attachment_size_and_component_termination_passage_based_land_phy
    SUBTYPE OF (default_passage_based_land_physical_template);
        SELF\default_passage_based_land_physical_template.of_passage_technology : default_co
END_ENTITY; -- default_component_termination_passage_based_land_physical_template

```

```

ENTITY default_component_termination_passage_definition
    SUBTYPE OF (passage_technology);
        minimum_allowed_component_terminal_extent : length_data_element;
        maximum_allowed_component_terminal_extent : length_data_element;
        SELF\passage_technology.deposition_material : ee_material;
        SELF\passage_technology.minimum_as_finished_deposition_thickness : length_data_element;
        SELF\passage_technology.minimum_as_finished_passage_extent : length_data_element;
END_ENTITY; -- default_component_termination_passage_definition

```

```

ENTITY default_land_physical_template
    ABSTRACT SUPERTYPE OF (ONEOF (default_passage_based_land_physical_template,
default_attachment_size_based_land_physical_template))
    SUBTYPE OF (land_physical_template);
        SELF\stratum_feature_template.of_stratum_technology : design_layer_technology;
END_ENTITY; -- default_land_physical_template

```

```

ENTITY default_passage_based_land_physical_template
  ABSTRACT SUPERTYPE OF (ONEOF (default_via_based_land_physical_template,
    default_component_termination_passage_based_land_physical_template,
    default_unsupported_passage_based_land_physical_template))
  SUBTYPE OF (default_land_physical_template);
    minimum_fabrication_allowance : length_data_element;
    of_passage_technology          : passage_technology;
    of_stratum_extent              : inter_stratum_extent;
    minimum_annular_ring          : length_data_element;
  UNIQUE
    UR1 : of_passage_technology,
  SELF\default_land_physical_template.of_stratum_technology;
END_ENTITY; -- default_passage_based_land_physical_template

```

```

ENTITY default_trace_template
  SUBTYPE OF (trace_template);
    of_stratum_technology : stratum_technology;
END_ENTITY; -- default_trace_template

```

```

ENTITY default_unsupported_passage_based_land_physical_template
  SUBTYPE OF (default_passage_based_land_physical_template);
    SELF\default_passage_based_land_physical_template.of_passage_technology : default_unsupported_passage_technology;
END_ENTITY; -- default_unsupported_passage_based_land_physical_template

```

```

ENTITY default_unsupported_passage_definition
  SUBTYPE OF (passage_technology);
    SELF\passage_technology.minimum_as_finished_passage_extent : length_data_element;
END_ENTITY; -- default_unsupported_passage_definition

```

```

ENTITY default_via_based_land_physical_template
  SUPERTYPE OF (default_attachment_size_and_via_based_land_physical_template)
  SUBTYPE OF (default_passage_based_land_physical_template);
    SELF\default_passage_based_land_physical_template.of_passage_technology : default_via_based_passage_technology;
END_ENTITY; -- default_via_based_land_physical_template

```

```

ENTITY default_via_definition
  SUBTYPE OF (passage_technology);
    SELF\passage_technology.deposition_material : ee_material;
    SELF\passage_technology.minimum_as_finished_deposition_thickness : length_data_element;
    SELF\passage_technology.minimum_as_finished_passage_extent : length_data_element;
END_ENTITY; -- default_via_definition

```

```

ENTITY delete_design_object_management_relationship

```



```

    SUBTYPE OF (design_object_management_relationship);
        previous_design_object : managed_design_object;
    UNIQUE
        UR1 : previous_design_object,
previous_design;
END_ENTITY; -- delete_design_object_management_relationship

```

```

ENTITY dependent_material_removal_feature_template
    SUBTYPE OF (material_removal_feature_template);
        associated_template : default_land_physical_template;
END_ENTITY; -- dependent_material_removal_feature_template

```

```

ENTITY dependently_located_layer_connection_point
    SUBTYPE OF (layer_connection_point);
        associated_design_object : terminal_or_inter_stratum_feature;
        reference_zone           : OPTIONAL connection_zone;
END_ENTITY; -- dependently_located_layer_connection_point

```

```

ENTITY derived_shape_element
    ABSTRACT SUPERTYPE OF (ONEOF (apex,
centre_of_symmetry,
linear_extension,
geometric_intersection,
geometric_alignment,
tangent_shape_element))
    SUBTYPE OF (non_feature_shape_element);
    INVERSE
        deriving_relationships : SET[1:?] OF shape_element_deriving_relationship FOR relating
END_ENTITY; -- derived_shape_element

```

```

ENTITY design_characteristic_occurrence
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- design_characteristic_occurrence

```

```

ENTITY design_composition_path
    SUBTYPE OF (managed_design_object);
        path_element           : SET[1:?] OF functional_unit;
        design_functional_unit : functional_unit;
END_ENTITY; -- design_composition_path

```

```

ENTITY design_functional_unit_allocation_to_assembly_component
    SUPERTYPE OF (transmission_line_functional_unit_to_routed_printed_component_allocation)
    SUBTYPE OF (managed_design_object);
        functional_design_definition_path : design_composition_path;

```

```

        implementation          : assembly_component;
        global_swappable       : BOOLEAN;
        local_swappable        : BOOLEAN;
        of_property             : OPTIONAL SET[1:?] OF coordinated_characteristic;
        path_alias              : ee_name;
        swap_code               : STRING;
    UNIQUE
        UR1 : implementation,
path_alias;
END_ENTITY; -- design_functional_unit_allocation_to_assembly_component

```

```

ENTITY design_functional_unit_allocation_to_assembly_joint
    SUBTYPE OF (managed_design_object);
        functional_design_definition_path : design_composition_path;
        implementation                    : assembly_joint;
    UNIQUE
        UR1 : functional_design_definition_path;
END_ENTITY; -- design_functional_unit_allocation_to_assembly_joint

```

```

ENTITY design_functional_unit_allocation_to_fabrication_joint
    SUBTYPE OF (managed_design_object);
        functional_design_definition_path : design_composition_path;
        implementation                    : fabrication_joint;
    UNIQUE
        UR1 : functional_design_definition_path;
END_ENTITY; -- design_functional_unit_allocation_to_fabrication_joint

```

```

ENTITY design_functional_unit_allocation_to_reference_functional_unit
    SUBTYPE OF (managed_design_object);
        reference_definition_path : reference_composition_path;
        design_definition_path    : design_composition_path;
END_ENTITY; -- design_functional_unit_allocation_to_reference_functional_unit

```

```

ENTITY design_intent_modification
    SUPERTYPE OF (ONEOF (material_removal_feature,
material_addition_feature,
material_shape_modification))
    SUBTYPE OF (managed_design_object);
        design_intent          : stratum_feature;
        modification_causal_item : restriction_basis_item;
END_ENTITY; -- design_intent_modification

```

```

ENTITY design_intent_modification_3d_position
    SUBTYPE OF (managed_design_object);
        design_shape : physical_unit_3d_shape;

```

```

        transformation : axis_placement_3d;
        feature_shape   : design_intent_modification_3d_shape_select;
    END_ENTITY; -- design_intent_modification_3d_position

```

```

ENTITY design_intent_modification_3d_shape
    SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
    represented_design_intent_modification : design_intent_modification;
    END_ENTITY; -- design_intent_modification_3d_shape

```

```

ENTITY design_intent_modification_non_planar_2d_shape
    SUBTYPE OF (open_shell_based_surface);
    represented_design_intent_modification : design_intent_modification;
    END_ENTITY; -- design_intent_modification_non_planar_2d_shape

```

```

ENTITY design_intent_modification_planar_position
    SUBTYPE OF (managed_design_object);
    design_shape      : physical_unit_planar_shape;
    transformation    : axis_placement_2d;
    feature_shape     : design_intent_modification_planar_shape_select;
    END_ENTITY; -- design_intent_modification_planar_position

```

```

ENTITY design_intent_modification_planar_shape
    SUBTYPE OF (planar_projected_shape);
    represented_design_intent_modification : design_intent_modification;
    END_ENTITY; -- design_intent_modification_planar_shape

```

```

ENTITY design_intent_modification_sequence
    SUBTYPE OF (managed_design_object);
    precedent_modification : design_intent_modification_or_material_removal_component;
    subsequent_modification : design_intent_modification_or_material_removal_component;
    UNIQUE
        UR1 : precedent_modification,
subsequent_modification;
    END_ENTITY; -- design_intent_modification_sequence

```

```

ENTITY design_intent_modification_styled_area_representation
    SUBTYPE OF (managed_design_object);
    derived_planar_shape          : OPTIONAL SET[1:?] OF design_intent_modification;
    represented_design_intent_modification : design_intent_modification;
    shape_styled_area_representation : styled_area_representation;
    END_ENTITY; -- design_intent_modification_styled_area_representation

```

```

ENTITY design_layer_stratum
  SUBTYPE OF (stratum);
    primary_design_layer_stratum : BOOLEAN;
    SELF\stratum.of_technology    : design_layer_technology;
  INVERSE
    existence_dependence : SET[1:?] OF layer_connection_point FOR resident_design_layer_
  WHERE
    WR1 : (NOT (primary_design_layer_stratum) AND (SIZEOF(QUERY(imps <* USEDIN(SELF,
'AP210_ARM.STRATUM_CONCEPT_TO_PHYSICAL_USAGE_VIEW_ASSIGNMENT.' + 'ASSIGNED_DESIGN_OBJECT'
'AP210_ARM.STRATUM_CONCEPT_TO_PHYSICAL_USAGE_VIEW_ASSIGNMENT.' + 'ASSIGNED_DESIGN_OBJECT'
END_ENTITY; -- design_layer_stratum

```

```

ENTITY design_layer_technology
  SUBTYPE OF (stratum_technology);
    design_layer_purpose           : layer_purpose;
    SELF\stratum_technology.minimum_finished_feature_size : length_data_element;
    SELF\stratum_technology.minimum_finished_feature_spacing : length_data_element;
END_ENTITY; -- design_layer_technology

```

```

ENTITY design_layer_type_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- design_layer_type_specific_padstack_definition

```

```

ENTITY design_object_management_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (change_design_object_management_relationship,
delete_design_object_management_relationship,
add_design_object_management_relationship));
    previous_design : ee_product_definition;
    current_design  : ee_product_definition;
END_ENTITY; -- design_object_management_relationship

```

```

ENTITY design_requirement_occurrence
  SUPERTYPE OF (interface_requirement_occurrence)
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- design_requirement_occurrence

```

```

ENTITY design_specification
  SUPERTYPE OF (interface_specification)
  SUBTYPE OF (ee_specification);
END_ENTITY; -- design_specification

```

```

ENTITY design_view_connection_zone_position_relationship
  SUBTYPE OF (managed_design_object);

```

```

    associating_design_view_shape          : feature_shape_definition;
    associated_usage                        : connection_zone;
    associated_connection_zone_shape_definition : feature_shape_definition;
    associated_usage_placement              : axis_placement;
    associating_design_view                 : physical_unit_design_view;
END_ENTITY; -- design_view_connection_zone_position_relationship

```

```

ENTITY design_view_terminal_component_shape_relationship
  SUBTYPE OF (managed_design_object);
    associating_component_shape          : feature_shape_definition;
    associated_usage                     : conductive_interconnect_terminal_or_interconn
    associated_terminal_shape_definition : feature_shape_definition;
    associated_usage_placement           : axis_placement;
END_ENTITY; -- design_view_terminal_component_shape_relationship

```

```

ENTITY device_2d_position
  SUBTYPE OF (managed_design_object);
    placed_device : part_device;
    transformation : axis_placement_2d;
    part_shape    : physical_unit_planar_shape;
    device_shape  : physical_unit_planar_shape;
END_ENTITY; -- device_2d_position

```

```

ENTITY device_3d_position
  SUBTYPE OF (managed_design_object);
    placed_device : part_device;
    transformation : axis_placement_3d;
    part_shape    : physical_unit_3d_shape;
    device_shape  : physical_unit_3d_shape;
END_ENTITY; -- device_3d_position

```

```

ENTITY device_terminal_map
  SUBTYPE OF (managed_design_object);
    target_terminal : packaged_part_terminal;
    mapped_device_terminal : part_device_terminal;
  UNIQUE
    UR1 : mapped_device_terminal,
target_terminal;
END_ENTITY; -- device_terminal_map

```

```

ENTITY device_terminal_map_relationship
  SUBTYPE OF (managed_design_object);
    related_device_terminal_map : device_terminal_map;
    relating_device_terminal_map : device_terminal_map;
END_ENTITY; -- device_terminal_map_relationship

```

```

ENTITY diametrical_boundary_size_characteristic
  SUBTYPE OF (boundary_size_characteristic);
    sized_boundary : tolerance_zone_circular_or_cylindrical_or_spherical_boundary;
END_ENTITY; -- diametrical_boundary_size_characteristic

```

```

ENTITY dielectric_crossover_area
  SUBTYPE OF (stratum_concept_relationship);
    related_concept      : stratum_feature;
    relating_concept     : stratum_feature;
    resulting_dielectric_crossover : stratum_feature;
END_ENTITY; -- dielectric_crossover_area

```

```

ENTITY dielectric_material_passage
  SUBTYPE OF (inter_stratum_feature);
    feature_material : OPTIONAL material_or_specification;
    precedent_passage : cutout;
END_ENTITY; -- dielectric_material_passage

```

```

ENTITY digital_analytical_model_port
  SUBTYPE OF (analytical_model_port);
  WHERE
    WR1 : not_applicable <> SELF\analytical_model_port.nominal_signal_flow_direction;
END_ENTITY; -- digital_analytical_model_port

```

```

ENTITY dimension_along_curve
  SUBTYPE OF (dimensional_location);
    path : shape_element;
  UNIQUE
    UR1 : located_elements,
path;
END_ENTITY; -- dimension_along_curve

```

```

ENTITY dimension_location_with_direction_vector
  SUBTYPE OF (dimensional_location);
    measurement_direction : axis_placement_or_datum_based_vector_orientation_select;
  UNIQUE
    UR1 : located_elements,
measurement_direction;
END_ENTITY; -- dimension_location_with_direction_vector

```

```

ENTITY dimension_location_with_specified_datum_system
  SUBTYPE OF (dimensional_location);
    referenced_datum_system_defining_conditions : datum_system_defining_conditions_witho
END_ENTITY; -- dimension_location_with_specified_datum_system

```

```

ENTITY dimension_qualifier
  ABSTRACT SUPERTYPE OF (ONEOF (pre_defined_dimension_qualifier,
user_defined_dimension_qualifier))
  SUBTYPE OF (managed_design_object);
    qualified_dimensions : SET[1:?] OF dimensional_characteristic;
END_ENTITY; -- dimension_qualifier

```

```

ENTITY dimension_related_positional_boundary
  SUBTYPE OF (positional_boundary);
    boundary_members      : SET[1:?] OF positional_boundary_member;
    constrained_physical_feature : physical_feature_or_part_template;
END_ENTITY; -- dimension_related_positional_boundary

```

```

ENTITY dimension_related_positional_boundary_definition
  SUBTYPE OF (positional_boundary_definition);
    SELF\positional_boundary_definition.defined_positional_boundary : dimension_related_
characterized_boundary_member_definitions      : SET[1:?] OF position
END_ENTITY; -- dimension_related_positional_boundary_definition

```

```

ENTITY dimension_set
  SUBTYPE OF (managed_design_object);
    associated_dimensions : SET[1:?] OF dimensional_characteristic;
END_ENTITY; -- dimension_set

```

```

ENTITY dimensional_characteristic
  ABSTRACT SUPERTYPE OF (ONEOF (dimensional_location,
dimensional_size) ANDOR
statistical_dimensional_tolerance)
  SUBTYPE OF (managed_design_object);
    dimension_type : dimensional_types;
END_ENTITY; -- dimensional_characteristic

```

```

ENTITY dimensional_characteristic_representation
  SUBTYPE OF (managed_design_object);
    representation      : shape_dimension_representation;
    dimension            : dimensional_location_or_dimensional_size;
    auxiliary_dimension_representation : BOOLEAN;
END_ENTITY; -- dimensional_characteristic_representation

```

```

ENTITY dimensional_location
  SUPERTYPE OF (dimension_location_with_specified_datum_system ANDOR
ONEOF (dimension_along_curve,
angular_dimension_with_direction_vector,

```

```

dimension_location_with_direction_vector))
    SUBTYPE OF (dimensional_characteristic);
        located_elements : shape_element_locating_relationship;
        directed          : BOOLEAN;
    END_ENTITY; -- dimensional_location

```

```

ENTITY dimensional_size
    ABSTRACT SUPERTYPE OF (ONEOF (dimensional_size_based_on_opposing_boundaries,
single_boundary_dimension) ANDOR
angular_size_dimension)
    SUBTYPE OF (dimensional_characteristic);
        size_of : shape_element;
        envelope : BOOLEAN;
    END_ENTITY; -- dimensional_size

```

```

ENTITY dimensional_size_based_on_opposing_boundaries
    SUBTYPE OF (dimensional_size);
        boundary_separation : shape_element_locating_relationship;
        SELF\dimensional_size.size_of : composite_unit_shape_element;
    END_ENTITY; -- dimensional_size_based_on_opposing_boundaries

```

```

ENTITY directed_axis
    SUBTYPE OF (non_feature_shape_element);
        axis_to_be_oriented : centre_axis;
        start_shape_element : shape_element;
        end_shape_element   : shape_element;
    UNIQUE
        UR1 : axis_to_be_oriented,
start_shape_element;
    END_ENTITY; -- directed_axis

```

```

ENTITY direction
    SUBTYPE OF (managed_design_object);
    END_ENTITY; -- direction

```

```

ENTITY direction_element
    SUBTYPE OF (managed_design_object);
        direction : datum_based_direction_vector;
        angle_value : angle_data_element;
        base_datum_plane : datum_plane;
        direction_of_half_datum_plane : datum_based_direction_vector;
    INVERSE
        elements_of : SET[1:?] OF datum_based_vector_orientation FOR direction_components;
    END_ENTITY; -- direction_element

```



```

ENTITY documentation_layer_stratum
  SUBTYPE OF (stratum);
  SELF\stratum.of_technology : documentation_layer_technology;
END_ENTITY; -- documentation_layer_stratum

```

```

ENTITY documentation_layer_technology
  SUBTYPE OF (stratum_technology);
  documentation_layer_purpose : STRING;
END_ENTITY; -- documentation_layer_technology

```

```

ENTITY dose_equivalent_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- dose_equivalent_data_element

```

```

ENTITY edge_based_2d_wireframe
  SUBTYPE OF (planar_shape);
END_ENTITY; -- edge_based_2d_wireframe

```

```

ENTITY edge_based_wireframe
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- edge_based_wireframe

```

```

ENTITY edge_segment_cross_section
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : edge_shape;
  SELF\volume_shape_intersection.projected_volume           : part_template_3d_shape;
  DERIVE
    cross_section_reference : edge_segment_vertex := SELF\volume_shape_intersection.proj
    associated_template      : inter_stratum_feature_edge_segment_template := SELF\volume
  INVERSE
    referencing_template : inter_stratum_feature_edge_segment_template FOR cross_section
  WHERE
    WR1 : associated_template :=: referencing_template;
END_ENTITY; -- edge_segment_cross_section

```

```

ENTITY edge_segment_vertex
  SUBTYPE OF (single_datum,
datum_point);
  UNIQUE
    UR1 : SELF\shape_element.name;
END_ENTITY; -- edge_segment_vertex

```

```

ENTITY edge_shape
  ABSTRACT SUPERTYPE OF (ONEOF (bevel_edge_feature_shape,
step_edge_feature_shape,
radius_edge_feature_shape))
  SUBTYPE OF (planar_shape,
feature_shape_definition);
  WHERE
    WR1 : SIZEOF(QUERY(escss <* USEDIN(SELF,
'AP210_ARM.VOLUME_SHAPE_INTERSECTION.SURFACE_INTERSECTION_RESULT') | NOT ('AP210_ARM.EDGE
END_ENTITY; -- edge_shape

```

```

ENTITY ee_approval
  SUBTYPE OF (managed_design_object);
  purpose : text;
  authorizer : SET[1:?] OF person_organization;
  approval_status : status;
  signoff_date : date_or_date_and_time;
END_ENTITY; -- ee_approval

```

```

ENTITY ee_colour
  SUBTYPE OF (managed_design_object);
  name : ee_name;
  UNIQUE
  UR1 : name;
END_ENTITY; -- ee_colour

```

```

ENTITY ee_document
  SUPERTYPE OF (ee_specification)
  SUBTYPE OF (managed_design_object);
  publisher : person_organization;
  key_phrase : OPTIONAL SET[1:?] OF structured_text;
  document_approval : OPTIONAL ee_approval;
  base_identifier : ee_document_identifier;
  change_date : OPTIONAL SET[1:?] OF date_and_time;
  creation_date : date_and_time;
  security_code : OPTIONAL security_classification;
  text_usage : OPTIONAL plain_or_structured_text;
  author : OPTIONAL SET[1:?] OF person;
  revision : ee_name;
  document_name : ee_name;
  UNIQUE
  UR1 : base_identifier,
revision;
END_ENTITY; -- ee_document

```

```

ENTITY ee_document_identifier

```

```

    SUBTYPE OF (managed_design_object);
        identifier_name : ee_name;
        assigned_value  : ee_name;
    UNIQUE
        UR1 : identifier_name,
assigned_value;
    END_ENTITY; -- ee_document_identifier

ENTITY ee_material
    SUBTYPE OF (managed_design_object);
        material_name           : STRING;
        electrical_conductivity_classification : OPTIONAL material_conductivity_classification;
        optical_conductivity_classification  : OPTIONAL material_conductivity_classification;
        registration_document       : ee_document;
        thermal_conductivity_classification : OPTIONAL material_conductivity_classification;
    UNIQUE
        UR1 : material_name,
registration_document;
    END_ENTITY; -- ee_material

ENTITY ee_product
    SUBTYPE OF (managed_design_object);
        description           : OPTIONAL text;
        product_nomenclature  : ee_name;
        part_number           : ee_name;
        design_owner          : person_organization;
        standard_product_indicator : OPTIONAL BOOLEAN;
    INVERSE
        product_version       : SET[1:?] OF ee_product_version FOR versioned_item;
        product_classification : SET[1:?] OF ee_product_related_product_category FOR of_product;
    UNIQUE
        UR1 : design_owner,
part_number;
    WHERE
        WR1 : assigned_product_category_correlation(product_version,
product_classification);
    END_ENTITY; -- ee_product

ENTITY ee_product_category
    SUPERTYPE OF (ee_product_related_product_category)
    SUBTYPE OF (managed_design_object);
        category_parameter       : OPTIONAL SET[1:?] OF model_parameter;
        category_specific_values : OPTIONAL SET[1:?] OF parameter_assignment;
        description               : OPTIONAL ee_name;
        name                     : ee_name;
        reference_document        : ee_document;
    UNIQUE
        UR1 : name,

```

```
reference_document;
END_ENTITY; -- ee_product_category
```

```
ENTITY ee_product_category_relationship
  SUBTYPE OF (managed_design_object);
  sub_category : ee_product_category;
  category      : ee_product_category;
END_ENTITY; -- ee_product_category_relationship
```

```
ENTITY ee_product_configuration
  SUBTYPE OF (managed_design_object);
  satisfying_product      : OPTIONAL SET[1:?] OF ee_product_version;
  configuration_manager    : person_organization;
  item_id                 : STRING;
  product_configuration_approval : ee_approval;
  model_of_product        : ee_product_model;
  phase_of_product        : STRING;
  UNIQUE
    UR1 : item_id,
satisfying_product;
END_ENTITY; -- ee_product_configuration
```

```
ENTITY ee_product_definition
  ABSTRACT SUPERTYPE OF (ONEOF (analytical_model_definition,
ee_rule_definition,
evaluation_ee_product_definition,
footprint_definition,
functional_unit_definition,
material_definition,
padstack_definition,
physical_unit,
physical_unit_network_definition,
requirement_definition,
stratum,
template_definition) ANDOR
externally_defined_ee_product_definition)
  SUBTYPE OF (managed_design_object);
  creator      : SET[1:?] OF person_organization;
  product_definition_id : ee_name;
  security_code : OPTIONAL security_classification;
  context_description : OPTIONAL ee_text;
  version         : ee_product_version;
  product_definition_approval : ee_approval;
  creation_date    : date_and_time;
  documentation    : OPTIONAL SET[1:?] OF ee_document;
  cad_filename     : OPTIONAL ee_name;
  discipline_id    : OPTIONAL ee_text;
  life_cycle_status : text;
```

```

    UNIQUE
      UR1 : product_definition_id,
version;
END_ENTITY; -- ee_product_definition

ENTITY ee_product_definition_based_constraint_occurrence
  SUPERTYPE OF (shape_and_ee_product_definition_based_constraint_occurrence)
  SUBTYPE OF (constraint_occurrence);
  constraining_part : ee_product_definition;
END_ENTITY; -- ee_product_definition_based_constraint_occurrence

ENTITY ee_product_model
  SUBTYPE OF (managed_design_object);
  model_name : ee_text;
  UNIQUE
    UR1 : model_name;
END_ENTITY; -- ee_product_model

ENTITY ee_product_related_product_category
  SUBTYPE OF (ee_product_category);
  of_product : SET[1:?] OF ee_product;
END_ENTITY; -- ee_product_related_product_category

ENTITY ee_product_specific_parameter_value_assignment
  SUBTYPE OF (managed_design_object);
  associated_product_category : ee_product_related_product_category;
  of_product                  : ee_product;
  assigned_parameter          : parameter_assignment;
END_ENTITY; -- ee_product_specific_parameter_value_assignment

ENTITY ee_product_version
  SUPERTYPE OF (ee_rule_version)
  SUBTYPE OF (managed_design_object);
  creator                : person_organization;
  authorizing_agreement  : OPTIONAL contract;
  revision_code          : ee_name;
  versioned_item         : ee_product;
  security_code          : OPTIONAL security_classification;
  product_version_approval : ee_approval;
  INVERSE
    product_definition : SET[1:?] OF ee_product_definition FOR version;
  UNIQUE
    UR1 : versioned_item,
revision_code;
END_ENTITY; -- ee_product_version

```

```

ENTITY ee_product_version_supersedence
  SUBTYPE OF (managed_design_object);
    superseded_product_version : ee_product_version;
    superseding_product_version : ee_product_version;
  UNIQUE
    UR1 : superseded_product_version,
superseding_product_version;
END_ENTITY; -- ee_product_version_supersedence

```

```

ENTITY ee_requirement_occurrence
  SUPERTYPE OF (((ONEOF (thermal_requirement_occurrence,
electromagnetic_requirement_occurrence,
electrical_requirement_occurrence,
mechanical_requirement_occurrence) ANDOR
ONEOF (design_requirement_occurrence,
inspection_requirement_occurrence,
test_requirement_occurrence,
purpose_occurrence,
constraint_occurrence,
design_characteristic_occurrence)) ANDOR
ONEOF (layout_spacing_requirement_occurrence,
item_restricted_requirement_occurrence)) ANDOR
requirement_group_occurrence)
  SUBTYPE OF (managed_design_object);
    associated_definition : requirement_definition;
    required_analytical_representation : OPTIONAL SET[1:?] OF analytical_representation;
    required_functional_specification : OPTIONAL SET[1:?] OF functional_specification;
    required_part : OPTIONAL SET[1:?] OF ee_product;
    required_coordinated_characteristic : OPTIONAL SET[1:?] OF coordinated_characteristic;
    requirement_specification : SET[1:?] OF ee_specification;
    required_characteristic : OPTIONAL SET[1:?] OF characteristic;
    required_material : OPTIONAL ee_material;
    description : OPTIONAL ee_text;
    name : ee_name;
  INVERSE
    hierarchy_context : SET[0:1] OF requirement_composition FOR element;
  UNIQUE
    UR1 : associated_definition;
END_ENTITY; -- ee_requirement_occurrence

```

```

ENTITY ee_rule_complex_clause
  SUBTYPE OF (managed_design_object);
    conjunctive_content : SET[1:?] OF ee_rule_simple_clause;
    identifier : STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- ee_rule_complex_clause

```

```

ENTITY ee_rule_conclusion_definition
  SUBTYPE OF (managed_design_object);
    rule_definition      : ee_rule_definition;
    identifier           : STRING;
    conjunctive_content  : SET[1:?] OF ee_rule_simple_clause;
  UNIQUE
    UR1 : rule_definition,
identifier;
END_ENTITY; -- ee_rule_conclusion_definition

```

```

ENTITY ee_rule_definition
  SUBTYPE OF (ee_product_definition);
    SELF\ee_product_definition.version : ee_rule_version;
  INVERSE
    premise      : ee_rule_premise_definition FOR rule_definition;
    conclusion    : ee_rule_conclusion_definition FOR rule_definition;
END_ENTITY; -- ee_rule_definition

```

```

ENTITY ee_rule_function
  SUBTYPE OF (managed_design_object);
    identifier           : STRING;
    parameter           : SET[1:?] OF model_parameter;
    reference_document   : ee_document;
    source_code         : ee_specification;
  UNIQUE
    UR1 : identifier,
reference_document;
END_ENTITY; -- ee_rule_function

```

```

ENTITY ee_rule_premise_definition
  SUBTYPE OF (managed_design_object);
    rule_definition      : ee_rule_definition;
    identifier           : STRING;
    disjunctive_content  : SET[1:?] OF ee_rule_complex_clause;
  UNIQUE
    UR1 : rule_definition,
identifier;
END_ENTITY; -- ee_rule_premise_definition

```

```

ENTITY ee_rule_simple_clause
  SUBTYPE OF (managed_design_object);
    rule_function : ee_rule_function;
    parameter     : SET[1:?] OF parameter_assignment;
    identifier     : STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- ee_rule_simple_clause

```

```

ENTITY ee_rule_version
  SUBTYPE OF (ee_product_version);
  INVERSE
    management_action : SET[1:?] OF rule_action FOR subject_rule;
END_ENTITY; -- ee_rule_version

```

```

ENTITY ee_specification
  SUPERTYPE OF (ONEOF (design_specification,
process_specification,
material_specification,
test_specification,
language_reference_manual))
  SUBTYPE OF (ee_document);
  source : person_organization;
END_ENTITY; -- ee_specification

```

```

ENTITY ee_text
  SUBTYPE OF (managed_design_object);
  content : STRING;
END_ENTITY; -- ee_text

```

```

ENTITY electric_charge_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- electric_charge_data_element

```

```

ENTITY electric_current_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- electric_current_data_element

```

```

ENTITY electrical_isolation_laminate_component
  SUBTYPE OF (filled_area_material_removal_laminate_component);
  SELF\assembly_component.usage_definition : electrical_isolation_removal_template;
END_ENTITY; -- electrical_isolation_laminate_component

```

```

ENTITY electrical_isolation_removal_structure
  SUBTYPE OF (filled_area_material_removal_structure);
  electrical_isolation_spacing_requirement : ee_requirement_occurrence;
END_ENTITY; -- electrical_isolation_removal_structure

```

```

ENTITY electrical_isolation_removal_template
  SUBTYPE OF (material_removal_feature_template);
  electrical_isolation_spacing_requirement : layout_spacing_requirement_occurrence;
END_ENTITY; -- electrical_isolation_removal_template

```



```

ENTITY electrical_network_definition
    SUBTYPE OF (functional_unit_network_definition);
END_ENTITY; -- electrical_network_definition

```

```

ENTITY electrical_requirement_occurrence
    SUPERTYPE OF (material_electrical_conductivity_requirement_occurrence)
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- electrical_requirement_occurrence

```

```

ENTITY electromagnetic_compatibility_requirement_allocation
    SUPERTYPE OF (shield_allocation)
    SUBTYPE OF (assembly_requirement_allocation);
END_ENTITY; -- electromagnetic_compatibility_requirement_allocation

```

```

ENTITY electromagnetic_requirement_occurrence
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- electromagnetic_requirement_occurrence

```

```

ENTITY electromotive_force_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- electromotive_force_data_element

```

```

ENTITY elliptic_area
    SUBTYPE OF (primitive_2d);
    position      : axis_placement_2d;
    semi_axis_1   : length_data_element;
    semi_axis_2   : length_data_element;
END_ENTITY; -- elliptic_area

```

```

ENTITY embedded_component_terminal
    SUBTYPE OF (laminate_component_join_terminal);
    SELF\component_feature.definition      : bare_die_terminal_or_packaged_part_term
    SELF\component_feature.associated_component : bare_die_component_or_packaged_componen
END_ENTITY; -- embedded_component_terminal

```

```

ENTITY energy_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- energy_data_element

```

```

ENTITY engineering_make_from

```

```

    SUBTYPE OF (managed_design_object);
        base_design      : ee_product_definition;
        resultant_design : ee_product_definition;
    END_ENTITY; -- engineering_make_from

ENTITY equivalent_functional_terminals_assignment
    SUBTYPE OF (managed_design_object);
        equivalent_terminals : SET[2:?] OF functional_unit_usage_view_terminal_definition;
        name                  : ee_name;
    END_ENTITY; -- equivalent_functional_terminals_assignment

ENTITY equivalent_functional_unit_definition_assignment
    SUBTYPE OF (managed_design_object);
        equivalent_functional_unit_definitions : SET[2:?] OF functional_unit_definition;
        name                                   : ee_name;
    END_ENTITY; -- equivalent_functional_unit_definition_assignment

ENTITY evaluated_characteristic
    SUPERTYPE OF (rule_based_evaluated_characteristic)
    SUBTYPE OF (managed_design_object);
        planned_product_data      : planned_characteristic;
        product_data_status       : evaluation_type;
        product_characterization_data : OPTIONAL SET[1:?] OF coordinated_characteristic;
        evaluation_result          : evaluation_result_value;
    END_ENTITY; -- evaluated_characteristic

ENTITY evaluation_ee_product_definition
    SUBTYPE OF (ee_product_definition);
        evaluated_product_definition : ee_product_definition;
        evaluation_data               : SET[1:?] OF evaluated_characteristic;
    WHERE
        WR1 : TYPEOF(SELF) <> TYPEOF(evaluated_product_definition);
    END_ENTITY; -- evaluation_ee_product_definition

ENTITY event_data_element
    SUBTYPE OF (data_element);
    END_ENTITY; -- event_data_element

ENTITY expanded_uncertainty
    SUBTYPE OF (standard_uncertainty);
        coverage_factor : REAL;
    END_ENTITY; -- expanded_uncertainty

```

```

ENTITY explicitly_located_layer_connection_point
  SUBTYPE OF (layer_connection_point);
END_ENTITY; -- explicitly_located_layer_connection_point

```

```

ENTITY external_definition
  SUBTYPE OF (managed_design_object);
  code          : STRING;
  code_value    : STRING;
  reference_document : standard_or_reference_document;
END_ENTITY; -- external_definition

```

```

ENTITY externally_defined_curve_font
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_curve_font

```

```

ENTITY externally_defined_dimension_representation
  SUBTYPE OF (shape_dimension_representation);
  definitions      : SET[1:?] OF external_definition;
  dimension_value : OPTIONAL data_element;
END_ENTITY; -- externally_defined_dimension_representation

```

```

ENTITY externally_defined_ee_product_definition
  SUBTYPE OF (ee_product_definition);
  source : ee_name;
END_ENTITY; -- externally_defined_ee_product_definition

```

```

ENTITY externally_defined_hatch_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_hatch_style

```

```

ENTITY externally_defined_tile
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_tile

```

```

ENTITY externally_defined_tile_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- externally_defined_tile_style

```

```

ENTITY fabrication_joint
  SUBTYPE OF (managed_design_object);

```

```

        fabrication_features_1      : fabricated_feature_select;
        fabrication_features_2      : fabricated_feature_select;
        auxiliary_joint_material     : OPTIONAL stratum_feature_template_component;
        associated_interconnect_module : interconnect_module;
END_ENTITY; -- fabrication_joint

```

```

ENTITY fabrication_technology_specification
    SUBTYPE OF (process_specification);
END_ENTITY; -- fabrication_technology_specification

```

```

ENTITY fabrication_thickness_requirement_occurrence
    SUBTYPE OF (inspection_requirement_occurrence);
    characterized_class      : metal_or_dielectric;
    characterized_class_range : range_characteristic;
    tolerance                : tolerance_characteristic;
END_ENTITY; -- fabrication_thickness_requirement_occurrence

```

```

ENTITY feature_group_relationship
    SUBTYPE OF (managed_design_object);
    relating_feature : part_group_feature;
    related_feature  : part_feature;
END_ENTITY; -- feature_group_relationship

```

```

ENTITY feature_shape_definition
    SUBTYPE OF (shape_definition);
END_ENTITY; -- feature_shape_definition

```

```

ENTITY fiducial
    SUBTYPE OF (laminare_component);
END_ENTITY; -- fiducial

```

```

ENTITY fiducial_part_feature
    SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- fiducial_part_feature

```

```

ENTITY fiducial_stratum_feature
    SUBTYPE OF (stratum_feature);
END_ENTITY; -- fiducial_stratum_feature

```

```

ENTITY fill_area
    SUBTYPE OF (managed_design_object);
    boundaries : SET[1:?] OF closed_curve;
END_ENTITY; -- fill_area

```

```

ENTITY fill_area_occurrence
  SUBTYPE OF (managed_design_object);
    style          : SET[1:?] OF fill_area_style;
    filled_area     : fill_area;
    fill_style_target : cartesian_point;
    boundary_style  : curve_style;
END_ENTITY; -- fill_area_occurrence

```

```

ENTITY fill_area_style
  SUBTYPE OF (managed_design_object);
    name          : STRING;
    fill_style     : fill_style_select;
  UNIQUE
    UR1 : name;
END_ENTITY; -- fill_area_style

```

```

ENTITY fill_area_style_hatching
  SUBTYPE OF (managed_design_object);
    hatch_line_appearance : curve_style;
    start_of_next_hatch_line : vector;
    point_of_reference_hatch_line : cartesian_point;
    pattern_start          : cartesian_point;
    hatch_line_angle       : angle_data_element;
END_ENTITY; -- fill_area_style_hatching

```

```

ENTITY fill_area_style_tile
  SUBTYPE OF (managed_design_object);
    first_direction_repeat_factor : vector;
    second_direction_repeat_factor : vector;
    tile                          : fill_area_style_tile_shape_select;
    tiling_scale                  : positive_ratio_measure;
END_ENTITY; -- fill_area_style_tile

```

```

ENTITY fill_area_style_tile_curve_with_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- fill_area_style_tile_curve_with_style

```

```

ENTITY fill_area_style_tile_solid_region
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- fill_area_style_tile_solid_region

```

```

ENTITY fill_area_style_tile_symbol_with_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- fill_area_style_tile_symbol_with_style

```

```

ENTITY filled_area_material_removal_laminate_component
  SUPERTYPE OF (ONEOF (electrical_isolation_laminate_component,
thermal_isolation_laminate_component))
  SUBTYPE OF (material_removal_laminate_component);
  reference_padstack : padstack_definition;
  WHERE
    WR1 : NOT ('AP210_ARM.CONDUCTOR' IN TYPEOF(SELF\material_removal_laminate_component.
END_ENTITY; -- filled_area_material_removal_laminate_component

```

```

ENTITY filled_area_material_removal_structure
  ABSTRACT SUPERTYPE OF (ONEOF (electrical_isolation_removal_structure,
thermal_isolation_removal_structure))
  SUBTYPE OF (material_removal_feature);
  SELF\design_intent_modification.design_intent : conductive_filled_area;
END_ENTITY; -- filled_area_material_removal_structure

```

```

ENTITY flatness_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- flatness_tolerance

```

```

ENTITY font_definition
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- font_definition

```

```

ENTITY footprint_definition
  SUBTYPE OF (ee_product_definition);
  reference_breakout : OPTIONAL SET[1:?] OF assembly_group_component;
  INVERSE
    padstack : SET[1:?] OF padstack_location_in_footprint_definition FOR footprint;
    shape : footprint_definition_shape FOR shape_characterized_footprint_definition;
  WHERE
    WR1 : SIZEOF(QUERY(prpc <* USEDIN(SELF\ee_product_definition.version.versioned_item,
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | (prpc.name = 'template mode
END_ENTITY; -- footprint_definition

```

```

ENTITY footprint_definition_shape
  SUBTYPE OF (planar_projected_shape);
  shape_characterized_footprint_definition : footprint_definition;
  shape_environment : OPTIONAL environment;
  shape_material_condition : material_condition;
END_ENTITY; -- footprint_definition_shape

```

```

ENTITY footprint_occurrence

```

```

SUBTYPE OF (assembly_group_component);
    SELF\assembly_component.usage_definition : footprint_definition;
WHERE
    WR1 : SIZEOF(USEDIN(SELF,
'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT')) <= 1;
    WR2 : NOT (SIZEOF(USEDIN(SELF,
'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT')) = 1) OR (SIZEOF(QUERY(
'AP210_ARM.COMPONENT_MAKE_FROM_RELATIONSHIP.RESULTANT_COMPONENT') | 'AP210_ARM.ASSEMBLY_G
    WR3 : SIZEOF(TYPEOF(SELF) * ['AP210_ARM.PRINTED_COMPONENT',
'AP210_ARM.LAMINATE_COMPONENT',
'AP210_ARM.PHYSICAL_COMPONENT']) = 0;
END_ENTITY; -- footprint_occurrence

```

```

ENTITY footprint_occurrence_sub_assembly_relationship
    SUBTYPE OF (managed_design_object);
        related_component          : laminate_component;
        relating_component         : footprint_occurrence;
        sub_assembly_reference_designation : STRING;
        terminal_location          : STRING;
    UNIQUE
        UR1 : sub_assembly_reference_designation,
relating_component,
terminal_location;
END_ENTITY; -- footprint_occurrence_sub_assembly_relationship

```

```

ENTITY force_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- force_data_element

```

```

ENTITY frequency_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- frequency_data_element

```

```

ENTITY frozen_assignment
    SUBTYPE OF (managed_design_object);
        frozen_item      : SET[1:?] OF frozen_design_object;
        assigned_approval : ee_approval;
END_ENTITY; -- frozen_assignment

```

```

ENTITY frozen_design_object
    SUBTYPE OF (managed_design_object);
END_ENTITY; -- frozen_design_object

```

```

ENTITY functional_specification

```

```

SUBTYPE OF (managed_design_object);
    associated_definition      : functional_specification_definition;
    reference_signal           : signal_specification;
    characterizing_functional_unit : functional_unit;
    characterizing_signal       : signal_specification;
    characteristic_category     : external_definition;
WHERE
    WR1 : characterizing_functional_unit = reference_signal.characterizing_functional_uni
    WR2 : characterizing_functional_unit = characterizing_signal.characterizing_function
    WR3 : characterizing_functional_unit.composed_network = associated_definition;
END_ENTITY; -- functional_specification

ENTITY functional_specification_definition
    SUBTYPE OF (functional_unit_network_definition);
    INVERSE
        composing_specification : SET[1:?] OF functional_specification FOR associated_defini
END_ENTITY; -- functional_specification_definition

ENTITY functional_terminal_group
    SUBTYPE OF (managed_design_object);
    name          : ee_name;
    group_description : STRING;
    group_type     : STRING;
    UNIQUE
    UR1 : name;
END_ENTITY; -- functional_terminal_group

ENTITY functional_terminal_group_assignment
    SUBTYPE OF (managed_design_object);
    composed_group          : functional_terminal_group;
    functional_usage_view_terminal : scalar_terminal_definition;
END_ENTITY; -- functional_terminal_group_assignment

ENTITY functional_unit
    SUBTYPE OF (managed_design_object);
    definition          : functional_unit_definition;
    reference_designation : ee_name;
    composed_network    : functional_unit_network_definition;
    functional_property  : OPTIONAL SET[1:?] OF parameter_assignment;
    INVERSE
        access_mechanisms : SET[1:?] OF functional_unit_terminal FOR accessed_functional_uni
    UNIQUE
    UR1 : reference_designation,
        composed_network;
END_ENTITY; -- functional_unit

```



```

ENTITY functional_unit_definition
  ABSTRACT SUPERTYPE OF (ONEOF (functional_unit_network_definition,
functional_unit_usage_view))
  SUBTYPE OF (ee_product_definition);
    analytical_functional_unit_representation : OPTIONAL SET[1:?] OF analytical_represen
END_ENTITY; -- functional_unit_definition

```

```

ENTITY functional_unit_network_definition
  SUPERTYPE OF (thermal_network_definition ANDOR
electrical_network_definition ANDOR
functional_specification_definition)
  SUBTYPE OF (functional_unit_definition);
    usage_view : functional_unit_usage_view;
END_ENTITY; -- functional_unit_network_definition

```

```

ENTITY functional_unit_network_node_definition
  SUBTYPE OF (managed_design_object);
    associated_functional_unit_definition : functional_unit_network_definition;
    functional_unit_network_node_name      : ee_name;
  DERIVE
    valid_conservative_node : BOOLEAN := ((SIZEOF(QUERY(vc <* USEDIN(SELF,
'AP210_ARM.BUS_ELEMENT_LINK.PRECEDENT_ELEMENT') | vc <> [])) + SIZEOF(QUERY(vc <* USEDIN(S
'AP210_ARM.BUS_ELEMENT_LINK.SUBSEQUENT_ELEMENT') | vc <> [])) + SIZEOF(QUERY(vc <* USEDIN
'AP210_ARM.FUNCTIONAL_UNIT_TERMINAL_NODE_ASSIGNMENT.COMPOSED_NODE') | vc <> [])) + SIZEOF
  INVERSE
    external_node_access : SET[0:1] OF functional_unit_network_terminal_definition_node_
  UNIQUE
    UR1 : functional_unit_network_node_name,
associated_functional_unit_definition;
END_ENTITY; -- functional_unit_network_node_definition

```

```

ENTITY functional_unit_network_terminal_definition_bus_assignment
  SUBTYPE OF (managed_design_object);
    connected_bus      : bus_structural_definition;
    connected_terminal : functional_unit_usage_view_terminal_definition;
END_ENTITY; -- functional_unit_network_terminal_definition_bus_assignment

```

```

ENTITY functional_unit_network_terminal_definition_node_assignment
  SUBTYPE OF (managed_design_object);
    composed_node      : functional_unit_network_node_definition;
    connected_terminal : functional_unit_usage_view_terminal_definition;
END_ENTITY; -- functional_unit_network_terminal_definition_node_assignment

```

```

ENTITY functional_unit_requirement_allocation
  SUBTYPE OF (managed_design_object);

```

```

        functional_requirement      : ee_requirement_occurrence;
        functional_unit_assignment : functional_unit;
END_ENTITY; -- functional_unit_requirement_allocation

```

```

ENTITY functional_unit_terminal
    SUBTYPE OF (managed_design_object);
    accessed_functional_unit : functional_unit;
    definition                : functional_unit_usage_view_terminal_definition;
    INVERSE
        bus_assignment  : SET[0:1] OF functional_unit_terminal_bus_assignment FOR connected_
        node_assignment : SET[0:1] OF functional_unit_terminal_node_assignment FOR connected_
END_ENTITY; -- functional_unit_terminal

```

```

ENTITY functional_unit_terminal_bus_assignment
    SUBTYPE OF (managed_design_object);
    connected_bus      : bus_structural_definition;
    connected_terminal : functional_unit_terminal;
END_ENTITY; -- functional_unit_terminal_bus_assignment

```

```

ENTITY functional_unit_terminal_node_assignment
    SUBTYPE OF (managed_design_object);
    composed_node      : functional_unit_network_node_definition;
    connected_terminal : functional_unit_terminal;
END_ENTITY; -- functional_unit_terminal_node_assignment

```

```

ENTITY functional_unit_usage_view
    SUBTYPE OF (functional_unit_definition);
    INVERSE
        access_mechanism : SET[1:?] OF functional_unit_usage_view_terminal_definition FOR assig
END_ENTITY; -- functional_unit_usage_view

```

```

ENTITY functional_unit_usage_view_terminal_definition
    SUPERTYPE OF (scalar_terminal_definition)
    SUBTYPE OF (managed_design_object);
    associated_functional_unit_definition : functional_unit_usage_view;
    signal_name                          : ee_name;
    INVERSE
        node_assignment : SET[0:?] OF functional_unit_network_terminal_definition_node_assignm
        bus_assignment  : SET[0:?] OF functional_unit_network_terminal_definition_bus_assignm
    UNIQUE
        UR1 : signal_name,
associated_functional_unit_definition;
    WHERE
        WR1 : (EXISTS(node_assignment) AND NOT (EXISTS(bus_assignment))) OR (NOT (EXISTS(node
END_ENTITY; -- functional_unit_usage_view_terminal_definition

```

```

ENTITY functional_unit_usage_view_terminal_definition_make_from_relationship
  SUBTYPE OF (managed_design_object);
    associated_make_from : engineering_make_from;
    reusable_feature      : functional_unit_usage_view_terminal_definition;
    resultant_feature     : functional_unit_usage_view_terminal_definition;
  UNIQUE
    UR1 : reusable_feature,
resultant_feature;
  WHERE
    WR1 : reusable_feature :<>: resultant_feature;
    WR2 : reusable_feature.associated_functional_unit_definition :<>: resultant_feature.
END_ENTITY; -- functional_unit_usage_view_terminal_definition_make_from_relationship

```

```

ENTITY functional_usage_view_to_part_terminal_assignment
  SUBTYPE OF (managed_design_object);
    physical_usage_view_terminal : part_terminal;
    functional_usage_view_terminal : scalar_terminal_definition;
  UNIQUE
    UR1 : functional_usage_view_terminal,
physical_usage_view_terminal;
END_ENTITY; -- functional_usage_view_to_part_terminal_assignment

```

```

ENTITY functional_usage_view_to_printed_part_template_terminal_assignment
  SUBTYPE OF (managed_design_object);
    physical_usage_view_terminal : printed_part_template_terminal;
    functional_usage_view_terminal : scalar_terminal_definition;
  UNIQUE
    UR1 : functional_usage_view_terminal,
physical_usage_view_terminal;
END_ENTITY; -- functional_usage_view_to_printed_part_template_terminal_assignment

```

```

ENTITY gate_path_swap_specification
  SUBTYPE OF (change_design_object_management_relationship);
    SELF\change_design_object_management_relationship.current_design_object : design_fur
    SELF\change_design_object_management_relationship.previous_design_object : design_fur
END_ENTITY; -- gate_path_swap_specification

```

```

ENTITY geometric_alignment
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- geometric_alignment

```

```

ENTITY geometric_intersection
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- geometric_intersection

```

```

ENTITY geometric_representation_relationship
    SUPERTYPE OF (ONEOF (shape_composition_relationship,
component_part_2d_geometric_representation_relationship,
component_part_2d_non_planar_geometric_representation_relationship,
component_part_3d_geometric_representation_relationship))
    SUBTYPE OF (managed_design_object);
    shape_1 : shape_definition;
    shape_2 : shape_definition;
    WHERE
        WR1 : shape_1 <> shape_2;
END_ENTITY; -- geometric_representation_relationship

```

```

ENTITY geometric_tolerance
    ABSTRACT SUPERTYPE OF (((ONEOF (geometric_tolerance_with_referenced_datum_system,
geometric_tolerance_without_referenced_datum_system) ANDOR
ONEOF (profile_of_any_line_tolerance,
position_tolerance,
profile_of_any_surface_tolerance)) ANDOR
geometric_tolerance_with_applied_material_condition_property) ANDOR
statistical_geometric_tolerance)
    SUBTYPE OF (managed_design_object);
    tolerance_zone_or_boundary_definition : constraining_element_definitions;
    toleranced_shape_element : toleranceable_shape_element;
END_ENTITY; -- geometric_tolerance

```

```

ENTITY geometric_tolerance_qualifier
    ABSTRACT SUPERTYPE OF (ONEOF (pre_defined_geometric_tolerance_qualifier,
user_defined_geometric_tolerance_qualifier))
    SUBTYPE OF (managed_design_object);
    qualified_geometric_tolerances : SET[1:?] OF geometric_tolerance;
END_ENTITY; -- geometric_tolerance_qualifier

```

```

ENTITY geometric_tolerance_with_applied_material_condition_property
    SUPERTYPE OF (boundary_based_position_tolerance)
    SUBTYPE OF (geometric_tolerance);
    applied_material_condition_property_name : material_condition_property_names;
END_ENTITY; -- geometric_tolerance_with_applied_material_condition_property

```

```

ENTITY geometric_tolerance_with_referenced_datum_system
    ABSTRACT SUPERTYPE OF (ONEOF (angularity_tolerance,
parallelism_tolerance,
perpendicularity_tolerance,
circular_runout_tolerance,
symmetry_tolerance,
concentricity_and_coaxiality_tolerance,
profile_of_any_line_tolerance_with_referenced_datum_system,

```

```

profile_of_any_surface_tolerance_with_referenced_datum_system,
position_tolerance_with_referenced_datum_system,
total_runout_tolerance))
    SUBTYPE OF (geometric_tolerance);
        datum_system_definition : datum_system_defining_conditions;
END_ENTITY; -- geometric_tolerance_with_referenced_datum_system

```

```

ENTITY geometric_tolerance_without_referenced_datum_system
    ABSTRACT SUPERTYPE OF (ONEOF (circularity_tolerance,
cylindricity_tolerance,
flatness_tolerance,
straightness_tolerance,
position_tolerance_without_referenced_datum_system,
profile_of_any_surface_tolerance_without_referenced_datum_system,
profile_of_any_line_tolerance_without_referenced_datum_system))
    SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- geometric_tolerance_without_referenced_datum_system

```

```

ENTITY geometrically_bounded_2d_wireframe
    SUBTYPE OF (planar_shape);
END_ENTITY; -- geometrically_bounded_2d_wireframe

```

```

ENTITY geometrically_bounded_surface
    SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- geometrically_bounded_surface

```

```

ENTITY guided_wave_bare_die_terminal
    SUBTYPE OF (minimally_defined_bare_die_terminal);
END_ENTITY; -- guided_wave_bare_die_terminal

```

```

ENTITY guided_wave_terminal
    SUBTYPE OF (package_terminal);
END_ENTITY; -- guided_wave_terminal

```

```

ENTITY illuminance_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- illuminance_data_element

```

```

ENTITY inductance_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- inductance_data_element

```

```

ENTITY inspection_requirement_occurrence
    SUPERTYPE OF (ONEOF (layout_line_width_tolerance_requirement_occurrence,
layout_land_width_tolerance_requirement_occurrence,
fabrication_thickness_requirement_occurrence))
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- inspection_requirement_occurrence

```

```

ENTITY inter_stratum_extent
    SUBTYPE OF (managed_design_object);
    from_stratum : stratum;
    to_stratum   : stratum;
END_ENTITY; -- inter_stratum_extent

```

```

ENTITY inter_stratum_feature
    ABSTRACT SUPERTYPE OF (ONEOF (unsupported_passage,
plated_inter_stratum_feature,
cutout,
interconnect_module_edge,
interconnect_module_edge_segment,
cutout_edge_segment,
dielectric_material_passage))
    SUBTYPE OF (laminate_component);
    feature_of_size : BOOLEAN;
    SELF\assembly_component.usage_definition : inter_stratum_feature_template;
END_ENTITY; -- inter_stratum_feature

```

```

ENTITY inter_stratum_feature_edge_segment_template
    SUBTYPE OF (inter_stratum_feature_template);
    composed_edge : inter_stratum_feature_edge_template;
    start_vertex  : edge_segment_vertex;
    end_vertex    : edge_segment_vertex;
    reference_path : trimmed_line;
    cross_section : OPTIONAL edge_segment_cross_section;
    WHERE
        WR1 : EXISTS(cross_section) XOR (SIZEOF(USEDIN(SELF\part_template,
'AP210_ARM.PART_TEMPLATE_3D_SHAPE.' + 'SHAPE_CHARACTERIZED_PART_TEMPLATE')) >= 0);
        WR2 : EXISTS(reference_path.start_point) AND EXISTS(reference_path.end_point);
        WR3 : NOT (EXISTS(start_vertex\datum_point.location_in_2d)) OR (reference_path.start
        WR4 : NOT (EXISTS(start_vertex\datum_point.location_in_3d)) OR (reference_path.start
        WR5 : NOT (EXISTS(end_vertex\datum_point.location_in_2d)) OR (reference_path.end_poi
        WR6 : NOT (EXISTS(end_vertex\datum_point.location_in_3d)) OR (reference_path.end_poi
        WR7 : start_vertex :<=: end_vertex;
END_ENTITY; -- inter_stratum_feature_edge_segment_template

```

```

ENTITY inter_stratum_feature_edge_template
    SUBTYPE OF (inter_stratum_feature_template);

```

```

    INVERSE
        edges : SET[2:?] OF inter_stratum_feature_edge_segment_template FOR composed_edge;
    END_ENTITY; -- inter_stratum_feature_edge_template

```

```

    ENTITY inter_stratum_feature_template
        SUPERTYPE OF (ONEOF (component_termination_passage_template,
        via_template,
        unsupported_passage_template,
        inter_stratum_feature_edge_segment_template,
        inter_stratum_feature_edge_template))
        SUBTYPE OF (part_template);
        of_passage_technology : passage_technology;
    END_ENTITY; -- inter_stratum_feature_template

```

```

    ENTITY inter_stratum_join_implementation
        ABSTRACT SUPERTYPE OF (ONEOF (physical_network_supporting_stratum_feature_conductive_j
        physical_network_supporting_inter_stratum_feature))
        SUBTYPE OF (managed_design_object);
        topological_requirement : inter_stratum_join_relationship;
    END_ENTITY; -- inter_stratum_join_implementation

```

```

    ENTITY inter_stratum_join_relationship
        SUBTYPE OF (join_relationship);
        points_to_be_connected : SET[2:?] OF layer_connection_point;
    END_ENTITY; -- inter_stratum_join_relationship

```

```

    ENTITY interconnect_component_interface_terminal
        SUBTYPE OF (component_interface_terminal);
        connection : OPTIONAL connection_zone;
        UNIQUE
            UR1 : SELF\shape_element.name,
            SELF\component_feature.associated_component;
    END_ENTITY; -- interconnect_component_interface_terminal

```

```

    ENTITY interconnect_component_join_terminal
        SUBTYPE OF (physical_component_terminal);
        connection : OPTIONAL connection_zone;
        SELF\component_feature.associated_component : physical_component;
        UNIQUE
            UR1 : SELF\shape_element.name,
            SELF\component_feature.associated_component;
    END_ENTITY; -- interconnect_component_join_terminal

```

```

    ENTITY interconnect_module

```

```

    SUPERTYPE OF (ONEOF (pcb,
reference_packaged_part_interconnect_implementation))
    SUBTYPE OF (physical_unit_design_view);
        assembly_design_requirement : OPTIONAL SET[1:?] OF assembly_module;
    INVERSE
        of_fabrication_thickness_requirement : SET[0:?] OF requirement_allocation FOR target;
    WHERE
        WR1 : SIZEOF(QUERY(ftr <* of_fabrication_thickness_requirement | NOT ('AP210_ARM.FAB
        WR2 : SIZEOF(QUERY(imsar <* USEDIN(SELF,
'AP210_ARM.INTERCONNECT_MODULE_STRATUM_ASSEMBLY_RELATIONSHIP.' + 'ASSEMBLY') | (('AP210_ARM
END_ENTITY; -- interconnect_module

```

```

ENTITY interconnect_module_cavity_surface
    SUBTYPE OF (interconnect_module_surface_feature);
        reference_surface : interconnect_module_surface_feature;
    WHERE
        WR1 : SELF :<>: reference_surface;
END_ENTITY; -- interconnect_module_cavity_surface

```

```

ENTITY interconnect_module_component
    SUPERTYPE OF (interconnect_module_macro_component)
    SUBTYPE OF (physical_component);
        SELF\assembly_component.usage_definition : interconnect_module_usage_view;
END_ENTITY; -- interconnect_module_component

```

```

ENTITY interconnect_module_component_stratum_based_terminal
    SUBTYPE OF (interconnect_module_component_terminal);
        SELF\component_feature.definition : interconnect_module_stratum_based_terminal;
END_ENTITY; -- interconnect_module_component_stratum_based_terminal

```

```

ENTITY interconnect_module_component_surface_feature
    SUBTYPE OF (physical_component_feature);
        SELF\component_feature.definition : interconnect_module_surface_feature;
END_ENTITY; -- interconnect_module_component_surface_feature

```

```

ENTITY interconnect_module_component_terminal
    SUPERTYPE OF ((interconnect_module_component_stratum_based_terminal) ANDOR
(interconnect_module_macro_component_join_terminal))
    SUBTYPE OF (physical_component_terminal);
        SELF\component_feature.definition : interconnect_module_terminal;
        SELF\component_feature.associated_component : interconnect_module_component;
END_ENTITY; -- interconnect_module_component_terminal

```

```

ENTITY interconnect_module_constraint_region

```



```

SUBTYPE OF (non_feature_shape_element);
    keepout : BOOLEAN;
    non_conformant_interconnect_module_design_object : OPTIONAL SET[1:?] OF interconnect_module_design_object;
    SELF\non_feature_shape_element.scope : interconnect_module;
    design_specific_purpose : ee_name;
    requirement : ee_requirement_occurrence;
    associated_stratum_extent : SET[1:?] OF inter_stratum_extent;
    constrained_design_object_category : interconnect_module_design_object;
INVERSE
    precedent_version_constraint_region : SET[0:1] OF non_feature_shape_element_relation;
WHERE
    WR1 : ('AP210_ARM.CSG_2D_SHAPE' IN TYPEOF(SELF\non_feature_shape_element.element_shape));
    WR2 : NOT (SIZEOF(precedent_version_constraint_region) = 1) OR ('AP210_ARM.INTERCONNECT_MODULE_CONSTRAINT_REGION' IN TYPEOF(precedent_version_constraint_region));
END_ENTITY; -- interconnect_module_constraint_region

```

```

ENTITY interconnect_module_cutout_segment_surface
    SUBTYPE OF (interconnect_module_surface_feature);
        composed_surface : interconnect_module_cutout_surface;
        start_vertex : edge_segment_vertex;
        end_vertex : edge_segment_vertex;
    WHERE
        WR1 : start_vertex :<>: end_vertex;
END_ENTITY; -- interconnect_module_cutout_segment_surface

```

```

ENTITY interconnect_module_cutout_surface
    SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_cutout_surface

```

```

ENTITY interconnect_module_edge
    SUPERTYPE OF (ONEOF (partially_plated_interconnect_module_edge,
        plated_interconnect_module_edge))
    SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- interconnect_module_edge

```

```

ENTITY interconnect_module_edge_segment
    SUPERTYPE OF (plated_interconnect_module_edge_segment)
    SUBTYPE OF (inter_stratum_feature);
        composed_edge : interconnect_module_edge;
        start_vertex : edge_segment_vertex;
        end_vertex : edge_segment_vertex;
        SELF\inter_stratum_feature.usage_definition : inter_stratum_feature_edge_segment_template;
    WHERE
        WR1 : start_vertex :<>: end_vertex;
END_ENTITY; -- interconnect_module_edge_segment

```

```

ENTITY interconnect_module_edge_segment_surface
  SUBTYPE OF (interconnect_module_surface_feature);
  composed_surface : interconnect_module_edge_surface;
  start_vertex      : edge_segment_vertex;
  end_vertex        : edge_segment_vertex;
  WHERE
    WR1 : start_vertex :<>: end_vertex;
END_ENTITY; -- interconnect_module_edge_segment_surface

```

```

ENTITY interconnect_module_edge_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_edge_surface

```

```

ENTITY interconnect_module_interface_terminal
  SUBTYPE OF (interconnect_module_terminal);
END_ENTITY; -- interconnect_module_interface_terminal

```

```

ENTITY interconnect_module_join_terminal
  SUBTYPE OF (interconnect_module_terminal);
  INVERSE
    defining_terminal : component_feature_to_physical_usage_view_assignment FOR defined_termin
  WHERE
    WR1 : SIZEOF(['AP210_ARM.COMPONENT_TERMINATION_PASSAGE_JOIN_TERMINAL',
'AP210_ARM.VIA_TERMINAL',
'AP210_ARM.PRINTED_COMPONENT_JOIN_TERMINAL',
'AP210_ARM.CONDUCTIVE_INTERCONNECT_ELEMENT_TERMINAL',
'AP210_ARM.LAND_JOIN_TERMINAL',
'AP210_ARM.NON_FUNCTIONAL_LAND_JOIN_TERMINAL'] * TYPEOF(defining_terminal.assigned_termin
END_ENTITY; -- interconnect_module_join_terminal

```

```

ENTITY interconnect_module_macro_component
  SUBTYPE OF (interconnect_module_component);
  design_definition : interconnect_module;
  INVERSE
    join_access_mechanisms : SET[1:?] OF interconnect_module_macro_component_join_termina
  WHERE
    WR1 : SIZEOF(QUERY(immcjt <* join_access_mechanisms | immcjt\component_feature.defin
END_ENTITY; -- interconnect_module_macro_component

```

```

ENTITY interconnect_module_macro_component_join_terminal
  SUBTYPE OF (interconnect_module_component_terminal);
  SELF\component_feature.definition : interconnect_module_join_terminal;
  SELF\component_feature.associated_component : interconnect_module_macro_component;
END_ENTITY; -- interconnect_module_macro_component_join_terminal

```

```

ENTITY interconnect_module_primary_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_primary_surface

```

```

ENTITY interconnect_module_product_assembly_view_fabrication_joint
  SUBTYPE OF (managed_design_object);
  joined_terminal : SET[1:?] OF interconnect_module_component_stratum_based_terminal;
END_ENTITY; -- interconnect_module_product_assembly_view_fabrication_joint

```

```

ENTITY interconnect_module_secondary_surface
  SUBTYPE OF (interconnect_module_surface_feature);
END_ENTITY; -- interconnect_module_secondary_surface

```

```

ENTITY interconnect_module_stratum_assembly_relationship
  SUBTYPE OF (managed_design_object);
  assembly          : interconnect_module;
  component         : stratum;
  reference_designation : ee_name;
  INVERSE
    stratum_planar_position : SET[0:1] OF stratum_interconnect_module_planar_position FOR stratum;
    stratum_3d_position     : SET[0:1] OF stratum_interconnect_module_3d_position FOR stratum;
  UNIQUE
    UR1 : assembly,
    reference_designation;
END_ENTITY; -- interconnect_module_stratum_assembly_relationship

```

```

ENTITY interconnect_module_stratum_based_terminal
  SUBTYPE OF (interconnect_module_terminal);
  WHERE
    WR1 : SIZEOF(QUERY(imtscr <* USEDIN(SELF,
'AP210_ARM.' + 'INTERCONNECT_MODULE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP.' + 'RELATE
'AP210_ARM.' + 'INTERCONNECT_MODULE_SECONDARY_SURFACE'] * TYPEOF(imtscr.relate
    WR2 : SIZEOF(USEDIN(SELF,
'AP210_ARM.' + 'INTERCONNECT_MODULE_TERMINAL_SURFACE_CONSTITUENT_RELATIONSHIP.' + 'RELATE
    WR3 : SIZEOF(['AP210_ARM.LAND',
'AP210_ARM.NON_FUNCTIONAL_LAND'] * TYPEOF(SELF\part_feature.design_view_definition_compon
END_ENTITY; -- interconnect_module_stratum_based_terminal

```

```

ENTITY interconnect_module_surface_feature
  ABSTRACT SUPERTYPE OF (ONEOF (interconnect_module_cavity_surface,
interconnect_module_cutout_segment_surface,
interconnect_module_cutout_surface,
interconnect_module_secondary_surface,
interconnect_module_primary_surface,

```

```

interconnect_module_edge_surface,
interconnect_module_edge_segment_surface))
    SUBTYPE OF (part_feature);
    SELF\part_feature.associated_definition : interconnect_module_usage_view;
END_ENTITY; -- interconnect_module_surface_feature

```

```

ENTITY interconnect_module_terminal
    SUPERTYPE OF (ONEOF (interconnect_module_interface_terminal,
interconnect_module_join_terminal))
    SUBTYPE OF (part_terminal);
    terminal_connection_zone : OPTIONAL SET[1:?] OF connection_zone;
    SELF\part_feature.associated_definition : interconnect_module_usage_view;
    INVERSE
    SELF\part_feature.feature_shape : SET[1:?] OF usage_concept_usage_relationship FOR as
END_ENTITY; -- interconnect_module_terminal

```

```

ENTITY interconnect_module_terminal_surface_constituent_relationship
    SUBTYPE OF (shape_element_constituent_relationship);
    SELF\shape_element_relationship.relate_shape_element : interconnect_module_surface
    SELF\shape_element_relationship.related_shape_element : interconnect_module_terminal
END_ENTITY; -- interconnect_module_terminal_surface_constituent_relationship

```

```

ENTITY interconnect_module_usage_view
    SUPERTYPE OF (pcb_usage_view)
    SUBTYPE OF (physical_unit_usage_view);
    minimum_thickness_over_metal_requirement : OPTIONAL length_data_element;
    maximum_thickness_over_metal_requirement : OPTIONAL length_data_element;
    minimum_thickness_over_dielectric_requirement : OPTIONAL length_data_element;
    maximum_thickness_over_dielectric_requirement : OPTIONAL length_data_element;
    measurement_condition : OPTIONAL restraint_condition;
    located_thickness_requirement : OPTIONAL SET[1:?] OF located_interconnect_usage_view;
    implemented_function : OPTIONAL functional_unit_usage_view;
    WHERE
        WR1 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' +
        WR2 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' +
        WR3 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE.ASSOCIATED_DEFINITION') | ('AP210_ARM.' +
        WR4 : EXISTS(maximum_thickness_over_metal_requirement) OR EXISTS(maximum_thickness_over_metal_requirement)
        WR5 : NOT (EXISTS(minimum_thickness_over_metal_requirement)) OR (minimum_thickness_over_metal_requirement)
        WR6 : NOT (EXISTS(maximum_thickness_over_metal_requirement)) OR (maximum_thickness_over_metal_requirement)
        WR7 : NOT (EXISTS(minimum_thickness_over_dielectric_requirement)) OR (minimum_thickness_over_dielectric_requirement)
        WR8 : NOT (EXISTS(maximum_thickness_over_dielectric_requirement)) OR (maximum_thickness_over_dielectric_requirement)
END_ENTITY; -- interconnect_module_usage_view

```

```

ENTITY interface_access_material_removal_laminate_component

```

```

    SUBTYPE OF (material_removal_laminate_component);
        accessed_interface : laminate_component_interface_terminal;
END_ENTITY; -- interface_access_material_removal_laminate_component

```

```

ENTITY interface_access_stratum_feature_template_component
    SUBTYPE OF (stratum_feature_template_component);
        accessed_interface : laminate_component_interface_terminal;
END_ENTITY; -- interface_access_stratum_feature_template_component

```

```

ENTITY interface_component
    SUPERTYPE OF (packaged_connector_component)
    SUBTYPE OF (physical_component);
    INVERSE
        access_mechanism : SET[1:?] OF component_interface_terminal FOR associated_component
END_ENTITY; -- interface_component

```

```

ENTITY interface_component_allocation
    SUBTYPE OF (managed_design_object);
        mating_solution      : interface_component;
        mating_requirement : mating_connector_component;
END_ENTITY; -- interface_component_allocation

```

```

ENTITY interface_mounted_join
    SUBTYPE OF (managed_design_object);
        connection_to_interconnect_module_component : packaged_connector_component_interface;
        access_to_interface_mounted_component       : physical_component_terminal;
END_ENTITY; -- interface_mounted_join

```

```

ENTITY interface_requirement_occurrence
    SUBTYPE OF (design_requirement_occurrence);
        interface_to_next_higher_assembly : next_higher_assembly_interface_requirement;
END_ENTITY; -- interface_requirement_occurrence

```

```

ENTITY interface_specification
    SUBTYPE OF (design_specification);
END_ENTITY; -- interface_specification

```

```

ENTITY interfaced_group_component
    SUBTYPE OF (assembly_group_component);
        group_interface_component : SET[1:?] OF interface_component;
END_ENTITY; -- interfaced_group_component

```

```

ENTITY interfacial_connection
  SUBTYPE OF (via);
END_ENTITY; -- interfacial_connection

```

```

ENTITY internal_probe_access_area
  SUBTYPE OF (probe_access_area);
  stratum_feature_implementation : stratum_feature;
END_ENTITY; -- internal_probe_access_area

```

```

ENTITY internal_stratum_access
  SUBTYPE OF (managed_design_object);
  connected_probe      : internal_probe_access_area;
  inter_stratum_feature : via;
  UNIQUE
    UR1 : connected_probe,
inter_stratum_feature;
END_ENTITY; -- internal_stratum_access

```

```

ENTITY intra_stratum_join_relationship
  SUBTYPE OF (join_relationship);
  points_to_be_connected : SET[2:?] OF layer_connection_point;
  INVERSE
    ordered_implementation : SET[0:1] OF conductor FOR implementation_requirement;
    random_implementation  : SET[0:1] OF connected_filled_area FOR implementation_requirement;
  WHERE
    WR1 : (EXISTS(ordered_implementation)) XOR (EXISTS(random_implementation));
END_ENTITY; -- intra_stratum_join_relationship

```

```

ENTITY item_restricted_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
  basis : restriction_basis_item;
END_ENTITY; -- item_restricted_requirement_occurrence

```

```

ENTITY join_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (inter_stratum_join_relationship,
intra_stratum_join_relationship,
unrouted_join_relationship))
  SUBTYPE OF (managed_design_object);
  associated_layout : interconnect_module;
  INVERSE
    network_topology : physical_network FOR interconnect_module_network_topology_element;
END_ENTITY; -- join_relationship

```

```

ENTITY join_two_physical_connectivity_definition_supporting_inter_stratum_feature
  SUBTYPE OF (physical_network_supporting_inter_stratum_feature);
  joined_definitions : SET[2:2] OF physical_connectivity_definition;
END_ENTITY; -- join_two_physical_connectivity_definition_supporting_inter_stratum_feature

```

```

ENTITY join_two_physical_connectivity_definition_supporting_printed_component
  SUBTYPE OF (conductive_interconnect_element_with_pre_defined_transitions);
  joined_definitions : SET[2:2] OF physical_connectivity_definition;
END_ENTITY; -- join_two_physical_connectivity_definition_supporting_printed_component

```

```

ENTITY junction_vertex_allocation
  SUBTYPE OF (managed_design_object);
  implementation_point : trace_junction_or_via;
  required_vertex      : topological_junction;
  UNIQUE
  UR1 : required_vertex;
END_ENTITY; -- junction_vertex_allocation

```

```

ENTITY laminate_component
  SUPERTYPE OF ((ONEOF (conductive_interconnect_element,
material_removal_laminate_component,
stratum_feature_template_component,
probe_access_area,
inter_stratum_feature,
special_symbol_laminate_component,
laminate_text_component,
laminate_text_string_component) ANDOR
fiducial) ANDOR
physical_laminate_component)
  SUBTYPE OF (assembly_component);
  cad_layer : OPTIONAL external_definition;
  SELF\assembly_component.usage_definition : OPTIONAL part_template;
END_ENTITY; -- laminate_component

```

```

ENTITY laminate_component_feature
  ABSTRACT SUPERTYPE OF (ONEOF (conductive_interconnect_element_terminal,
laminate_component_interface_terminal,
laminate_component_join_terminal,
via_terminal))
  SUBTYPE OF (component_feature);
END_ENTITY; -- laminate_component_feature

```

```

ENTITY laminate_component_interface_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (non_functional_land_interface_terminal,
printed_connector_component_interface_terminal,

```

```

component_termination_passage_interface_terminal,
land_interface_terminal))
    SUBTYPE OF (laminate_component_feature);
END_ENTITY; -- laminate_component_interface_terminal

```

```

ENTITY laminate_component_join_terminal
    ABSTRACT SUPERTYPE OF (ONEOF (component_termination_passage_join_terminal,
land_join_terminal,
non_functional_land_join_terminal,
embedded_component_terminal,
printed_component_join_terminal))
    SUBTYPE OF (laminate_component_feature,
component_terminal);
    INVERSE
        node_implementation : SET[0:1] OF physical_network FOR connectivity_requirement_element;
END_ENTITY; -- laminate_component_join_terminal

```

```

ENTITY laminate_group_component_make_from_relationship
    SUBTYPE OF (component_make_from_relationship);
    SELF\component_make_from_relationship.reusable_component : assembly_group_component;
    SELF\component_make_from_relationship.resultant_component : assembly_group_component;
    stratum_mapping : SET[1:?] OF stratum_make;
    WHERE
        WR1 : SIZEOF(QUERY(sar <* SELF\component_make_from_relationship.reusable_component.a
        WR2 : SIZEOF(QUERY(sar <* SELF\component_make_from_relationship.resultant_component.a
END_ENTITY; -- laminate_group_component_make_from_relationship

```

```

ENTITY laminate_text_component
    SUBTYPE OF (laminate_component);
    of_character : STRING (1) FIXED;
    SELF\assembly_component.usage_definition : part_text_template;
    stratum_feature_implementation : SET[1:?] OF stratum_feature;
END_ENTITY; -- laminate_text_component

```

```

ENTITY laminate_text_string_component
    SUBTYPE OF (array_placement_group,
laminate_component);
    message : message_content;
    of_horizontal_justification : horizontal_justification;
    of_vertical_justification : vertical_justification;
END_ENTITY; -- laminate_text_string_component

```

```

ENTITY land
    SUPERTYPE OF (ONEOF (via_dependent_land,
component_termination_passage_dependent_land,

```



```

contact_size_dependent_land,
component_termination_passage_and_contact_size_dependent_land,
via_and_contact_size_dependent_land))
    SUBTYPE OF (stratum_feature_template_component);
        alternate_land_definition          : OPTIONAL land_physical_template;
        SELF\assembly_component.usage_definition : OPTIONAL land_physical_template;
    INVERSE
        access_mechanism : SET[1:?] OF land_join_terminal FOR associated_component;
END_ENTITY; -- land

ENTITY land_interface_terminal
    SUBTYPE OF (laminate_component_interface_terminal);
        SELF\component_feature.definition          : land_template_interface_terminal;
        SELF\component_feature.associated_component : land;
    WHERE
        WR1 : SIZEOF(['AP210_ARM.CONTACT_SIZE_DEPENDENT_LAND',
'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_LAND',
'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND'] * TYPEOF(SELF\component_feature.associat
END_ENTITY; -- land_interface_terminal

ENTITY land_join_terminal
    SUBTYPE OF (laminate_component_join_terminal);
        SELF\component_feature.definition          : land_template_join_terminal;
        SELF\component_feature.associated_component : land;
END_ENTITY; -- land_join_terminal

ENTITY land_physical_template
    SUPERTYPE OF (default_land_physical_template)
    SUBTYPE OF (stratum_feature_template);
        pre_defined_classification_code : OPTIONAL external_definition;
    INVERSE
        access_mechanisms : SET[1:?] OF land_template_terminal FOR associated_definition;
    UNIQUE
        UR1 : pre_defined_classification_code;
    WHERE
        WR1 : SIZEOF(QUERY(ltjt <* access_mechanisms | NOT ('AP210_ARM.LAND_TEMPLATE_JOIN_TE
END_ENTITY; -- land_physical_template

ENTITY land_template_interface_terminal
    SUBTYPE OF (land_template_terminal);
        SELF\land_template_terminal.associated_definition : default_attachment_size_based_la
END_ENTITY; -- land_template_interface_terminal

ENTITY land_template_join_terminal
    SUBTYPE OF (land_template_terminal);
END_ENTITY; -- land_template_join_terminal

```

```

ENTITY land_template_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (land_template_join_terminal,
land_template_interface_terminal))
  SUBTYPE OF (shape_element);
    associated_definition      : land_physical_template;
    terminal_connection_zone   : OPTIONAL SET[1:?] OF connection_zone;
    SELF\shape_element.name    : ee_name;
    connection_zone_category   : OPTIONAL SET[1:?] OF land_template_terminal_class;
  INVERSE
    terminal_shape : SET[0:1] OF usage_concept_usage_relationship FOR associated_usage;
  UNIQUE
    UR1 : name,
associated_definition;
  WHERE
    WR1 : NOT (EXISTS(terminal_connection_zone)) OR EXISTS(connection_zone_category);
    WR2 : NOT (EXISTS(terminal_connection_zone)) OR (SIZEOF(terminal_connection_zone) = 0);
END_ENTITY; -- land_template_terminal

```

```

ENTITY language_reference_manual
  SUBTYPE OF (ee_specification);
END_ENTITY; -- language_reference_manual

```

```

ENTITY layer
  SUBTYPE OF (managed_design_object);
    extent      : inter_stratum_extent;
    layer_usage : OPTIONAL external_definition;
    name        : ee_name;
  UNIQUE
    UR1 : name;
END_ENTITY; -- layer

```

```

ENTITY layer_connection_point
  ABSTRACT SUPERTYPE OF (ONEOF (explicitly_located_layer_connection_point,
dependently_located_layer_connection_point))
  SUBTYPE OF (managed_design_object);
    location_2d      : OPTIONAL cartesian_point;
    location_3d      : OPTIONAL cartesian_point;
    name             : ee_name;
    resident_design_layer_stratum : design_layer_stratum;
  DERIVE
    referencing_network : SET OF physical_network := QUERY(ijp <* inter_join_path | SIZEOF(
'AP210_ARM.PHYSICAL_NETWORK.' + 'INTERCONNECT_MODULE_NETWORK_TOPOLOGY_ELEMENT') + USEDIN(
'AP210_ARM.PHYSICAL_NETWORK.' + 'INTERCONNECT_MODULE_NETWORK_TOPOLOGY_ELEMENT')) > 0);
  INVERSE
    inter_join_path : SET[0:1] OF inter_stratum_join_relationship FOR points_to_be_connected;
    intra_join_path : intra_stratum_join_relationship FOR points_to_be_connected;
  UNIQUE

```

```

        UR1 : name,
resident_design_layer_stratum;
WHERE
    WR1 : EXISTS(location_2d) OR EXISTS(location_3d);
    WR2 : NOT (EXISTS(location_2d)) OR (location_2d\shape_item.geometric_context.dimensions > 0);
    WR3 : NOT (EXISTS(location_2d)) OR (location_3d\shape_item.geometric_context.dimensions > 0);
    WR4 : SIZEOF(referencing_network) <= 1;
END_ENTITY; -- layer_connection_point

```

```

ENTITY layer_connection_point_link
    SUBTYPE OF (managed_design_object);
    associated_stratum : stratum;
    precedent_point    : layer_connection_point;
    subsequent_point   : layer_connection_point;
    INVERSE
        referenced_by : unrouted_conductive_interconnect_element FOR connected_points;
    WHERE
        WR1 : precedent_point <> subsequent_point;
        WR2 : associated_stratum = precedent_point.resident_design_layer_stratum;
        WR3 : associated_stratum = subsequent_point.resident_design_layer_stratum;
END_ENTITY; -- layer_connection_point_link

```

```

ENTITY layer_qualified_layout_spacing_requirement_occurrence
    SUBTYPE OF (layout_spacing_requirement_occurrence);
    layer_context : SET[1:?] OF inter_stratum_extent;
END_ENTITY; -- layer_qualified_layout_spacing_requirement_occurrence

```

```

ENTITY layout_land_width_tolerance_requirement_occurrence
    SUBTYPE OF (inspection_requirement_occurrence);
    mating_feature_pitch_class : pitch_class;
    maximum_negative_deviation : length_data_element;
END_ENTITY; -- layout_land_width_tolerance_requirement_occurrence

```

```

ENTITY layout_line_width_tolerance_requirement_occurrence
    SUBTYPE OF (inspection_requirement_occurrence);
    characterized_stratum_technology : stratum_technology;
    tolerance_line_width_class       : line_width_class;
    maximum_negative_deviation       : length_data_element;
    maximum_positive_deviation       : length_data_element;
END_ENTITY; -- layout_line_width_tolerance_requirement_occurrence

```

```

ENTITY layout_macro_component
    SUBTYPE OF (printed_component);
    design_definition : layout_macro_definition;
    SELF\printed_component.usage_definition : layout_macro_floor_plan_template;

```

```

WHERE
    WR1 : NOT (EXISTS(SELF\printed_component.stratum_feature_implementation)) AND NOT (EXISTS(SELF\interconnect_module.assembly_design_requirement));
END_ENTITY; -- layout_macro_component

```

```

ENTITY layout_macro_definition
    SUBTYPE OF (pcb);
    floor_layout_view : layout_macro_floor_plan_template;
    WHERE
        WR1 : NOT (EXISTS(SELF\physical_unit_design_view.usage_view));
        WR2 : NOT (EXISTS(SELF\interconnect_module.assembly_design_requirement));
        WR3 : SIZEOF(USEDIN(SELF,
'AP210_ARM.ASSEMBLY_COMPONENT.USAGE_DEFINITION')) = 0;
END_ENTITY; -- layout_macro_definition

```

```

ENTITY layout_macro_definition_terminal_to_usage_terminal_assignment
    SUBTYPE OF (managed_design_object);
    defined_template_feature : printed_part_template_terminal;
    assigned_design_object : fabricated_feature_select;
    WHERE
        WR1 : SIZEOF(USEDIN(SELF,
'AP210_ARM.LAYOUT_MACRO_FLOOR_PLAN_TEMPLATE.ACCESS_MAPS')) = 1;
END_ENTITY; -- layout_macro_definition_terminal_to_usage_terminal_assignment

```

```

ENTITY layout_macro_floor_plan_template
    SUBTYPE OF (printed_part_template);
    access_maps : SET[1:?] OF layout_macro_definition_terminal_to_usage_terminal_assignment;
    INVERSE
        macro_definition : SET[0:1] OF layout_macro_definition FOR floor_layout_view;
    WHERE
        WR1 : NOT (EXISTS(SELF\printed_part_template.of_stratum_technology)) AND NOT (EXISTS(SELF\interconnect_module.assembly_design_requirement));
END_ENTITY; -- layout_macro_floor_plan_template

```

```

ENTITY layout_spacing_requirement_non_conforming_design_object
    SUBTYPE OF (managed_design_object);
    non_conformant_design_object : SET[1:?] OF assembly_or_interconnect_design_object_select;
    reference_design_object : assembly_or_interconnect_design_object_select;
    design_requirement : layout_spacing_requirement_occurrence;
END_ENTITY; -- layout_spacing_requirement_non_conforming_design_object

```

```

ENTITY layout_spacing_requirement_occurrence
    SUPERTYPE OF (layer_qualified_layout_spacing_requirement_occurrence)
    SUBTYPE OF (ee_requirement_occurrence);
    reference_design_object_category : pca_product_design_category;
    dependent_design_object_category : pca_product_design_category;
    SELF\ee_requirement_occurrence.required_coordinated_characteristic : tolerance_characteristic;

```

```

        of_spacing_type                                     : spacing_type;
    UNIQUE
        UR1 : reference_design_object_category,
dependent_design_object_category;
    WHERE
        WR1 : 'AP210_ARM.LENGTH_DATA_ELEMENT' IN TYPEOF(SELF\ee_requirement_occurrence.requirement_occurrence);
END_ENTITY; -- layout_spacing_requirement_occurrence

ENTITY lead_form_specification
    SUBTYPE OF (process_specification);
END_ENTITY; -- lead_form_specification

ENTITY length_data_element
    SUBTYPE OF (data_element);
    WHERE
        WR1 : SELF\data_element.measure_value >= 0.0;
END_ENTITY; -- length_data_element

ENTITY length_trimmed_terminal
    SUBTYPE OF (altered_package_terminal);
    maximum_trimmed_length : length_data_element;
    minimum_trimmed_length : length_data_element;
END_ENTITY; -- length_trimmed_terminal

ENTITY library_ee_product_definition
    SUBTYPE OF (externally_defined_ee_product_definition);
    DERIVE
        library : ee_name := SELF\externally_defined_ee_product_definition.source;
END_ENTITY; -- library_ee_product_definition

ENTITY limit_dimension_representation
    SUBTYPE OF (shape_dimension_representation);
    upper_dimension_value : data_element;
    lower_dimension_value : data_element;
END_ENTITY; -- limit_dimension_representation

ENTITY line_width_class
    SUBTYPE OF (managed_design_object);
    class_member : range_characteristic;
    class_name   : ee_name;
    WHERE
        WR1 : SIZEOF(QUERY(np <* class_member\coordinated_characteristic.component_characteristic)) > 0;
END_ENTITY; -- line_width_class

```

```

ENTITY linear_array_placement_group_component
  SUBTYPE OF (array_placement_group);
    element : SET[1:?] OF assembly_component_placement_link;
    initial_node_positive_x_displacement : length_data_element;
    terminus_node_positive_x_displacement : length_data_element;
    location_uncertainty : length_data_element;
END_ENTITY; -- linear_array_placement_group_component

```

```

ENTITY linear_array_placement_group_component_link
  SUBTYPE OF (managed_design_object);
    precedent_element : linear_array_placement_group_component;
    subsequent_element : linear_array_placement_group_component;
  INVERSE
    referenced_by : rectangular_array_placement_group_component FOR element;
END_ENTITY; -- linear_array_placement_group_component_link

```

```

ENTITY linear_extension
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- linear_extension

```

```

ENTITY located_interconnect_module_thickness_requirement
  SUBTYPE OF (managed_design_object);
    reference_location : axis_placement;
    maximum_thickness_requirement : length_data_element;
    minimum_thickness_requirement : length_data_element;
  INVERSE
    design : SET[1:?] OF interconnect_module_usage_view FOR located_thickness_requirement;
  UNIQUE
    UR1 : design,
reference_location;
END_ENTITY; -- located_interconnect_module_thickness_requirement

```

```

ENTITY luminous_flux_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- luminous_flux_data_element

```

```

ENTITY luminous_intensity_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- luminous_intensity_data_element

```

```

ENTITY magnetic_flux_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- magnetic_flux_data_element

```

```

ENTITY magnetic_flux_density_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- magnetic_flux_density_data_element

```

```

ENTITY managed_design_object
  ABSTRACT SUPERTYPE OF ((ONEOF (adjacent_stratum_surface_definition,
adjacent_stratum_surface_embedded_component_surface_definition,
alternate_select_product,
analytical_model,
analytical_model_make_from_relationship,
analytical_model_port,
analytical_model_port_assignment,
analytical_representation,
assembly_bond_definition,
assembly_component,
assembly_component_placement_link,
assembly_composition_relationship,
assembly_joint,
assembly_material_composition_relationship,
assembly_requirement_allocation,
bond_assembly_position,
boundary_size_characteristic,
bus_element_link,
bus_structural_definition,
cartesian_coordinate_system,
cartesian_transformation_operator_2d,
certification,
characteristic,
component_2d_location,
component_3d_location,
component_assembly_2d_position,
component_assembly_3d_position,
component_assembly_non_planar_position,
component_external_reference,
component_feature,
component_feature_external_reference,
component_feature_make_from_relationship,
component_feature_relationship,
component_feature_to_layout_feature_relationship,
component_feature_to_physical_usage_view_assignment,
component_group_assignment,
component_make_from_relationship,
component_mounting_clearance_relationship,
component_non_planar_location,
component_overlap_relationship,
component_placement_restriction_assignment,
composite_curve_segment,
composite_signal_property_relationship,
conductive_interconnect_element_terminal_link,
configuration_controlled_message,

```

conical_tolerance_zone_boundary_and_surface_relationship,
conical_tolerance_zone_boundary_diametrical_size_characteristic,
connection_zone,
connectivity_allocation,
connectivity_allocation_to_physical_network,
connection_zone_interface_plane_relationship,
connection_zone_map_identification,
connection_zone_model_port_assignment,
coordinated_characteristic,
csg_primitive,
curve_style,
curve_style_font,
curve_style_font_and_scaling,
data_element,
date,
datum_axis_and_tolerance_zone_orienting_relationship,
datum_axis_related_orientation,
datum_based_direction_vector,
datum_based_length_measure,
datum_based_vector_orientation,
datum_difference_analytical_model_port_assignment,
datum_difference_functional_unit_usage_view_terminal_assignment,
datum_feature_material_condition_property,
datum_feature_usage_in_datum_system_relationship,
datum_precedence_assignment,
datum_system_defining_conditions,
datum_target_usage_in_datum_target_set_relationship,
datum_usage_in_datum_system_relationship,
design_composition_path,
design_functional_unit_allocation_to_assembly_component,
design_functional_unit_allocation_to_assembly_joint,
design_functional_unit_allocation_to_fabrication_joint,
design_functional_unit_allocation_to_reference_functional_unit,
design_intent_modification,
design_intent_modification_sequence,
design_intent_modification_3d_position,
design_intent_modification_planar_position,
design_intent_modification_styled_area_representation,
design_view_connection_zone_position_relationship,
design_view_terminal_component_shape_relationship,
device_2d_position,
device_3d_position,
device_terminal_map,
device_terminal_map_relationship,
dimensional_characteristic,
dimensional_characteristic_representation,
dimension_qualifier,
dimension_set,
direction,
direction_element,
ee_approval,
ee_colour,

ee_document,
ee_document_identifier,
ee_material,
ee_product,
ee_product_category,
ee_product_category_relationship,
ee_product_configuration,
ee_product_definition,
ee_product_model,
ee_product_specific_parameter_value_assignment,
ee_product_version,
ee_requirement_occurrence,
ee_rule_complex_clause,
ee_rule_conclusion_definition,
ee_rule_function,
ee_rule_premise_definition,
ee_rule_simple_clause,
ee_text,
engineering_make_from,
equivalent_functional_terminals_assignment,
equivalent_functional_unit_definition_assignment,
evaluated_characteristic,
externally_defined_curve_font,
externally_defined_hatch_style,
externally_defined_tile,
externally_defined_tile_style,
external_definition,
fabrication_joint,
feature_group_relationship,
fill_area,
fill_area_occurrence,
fill_area_style,
fill_area_style_hatching,
fill_area_style_tile,
fill_area_style_tile_curve_with_style,
fill_area_style_tile_solid_region,
fill_area_style_tile_symbol_with_style,
font_definition,
frozen_assignment,
functional_specification,
functional_terminal_group,
functional_terminal_group_assignment,
functional_unit,
functional_unit_network_node_definition,
functional_unit_network_terminal_definition_bus_assignment,
functional_unit_network_terminal_definition_node_assignment,
functional_unit_requirement_allocation,
functional_unit_terminal,
functional_unit_terminal_bus_assignment,
functional_unit_terminal_node_assignment,
functional_unit_usage_view_terminal_definition,
functional_usage_view_to_part_terminal_assignment,

functional_usage_view_to_printed_part_template_terminal_assignment,
 geometric_tolerance,
 geometric_tolerance_qualifier,
 interconnect_module_product_assembly_view_fabrication_joint,
 interconnect_module_stratum_assembly_relationship,
 interface_component_allocation,
 interface_mounted_join,
 internal_stratum_access,
 inter_stratum_extent,
 inter_stratum_join_implementation,
 join_relationship,
 junction_vertex_allocation,
 layer,
 layer_connection_point,
 layer_connection_point_link,
 layout_spacing_requirement_non_conforming_design_object,
 line_width_class,
 linear_array_placement_group_component_link,
 located_interconnect_module_thickness_requirement,
 marking,
 material_composition_relationship,
 mating_connector_component,
 mating_connector_placement_relationship,
 mating_connector_termination,
 model_parameter,
 next_higher_assembly_interface_requirement,
 operational_requirement_occurrence_relationship,
 organization,
 package_external_reference,
 package_terminal_external_reference,
 package_terminal_occurrence,
 packaged_connector_terminal_relationship,
 parameter_assignment,
 participant_date_and_time_assignment,
 part_connected_terminals_definition,
 part_device,
 part_device_terminal,
 part_device_terminal_model_port_assignment,
 part_feature_model_port_assignment,
 part_terminal_external_reference,
 part_template_or_physical_unit_2d_shape_representation_relationship,
 part_template_or_physical_unit_3d_shape_representation_relationship,
 passage_technology,
 person,
 person_and_organization,
 physical_connectivity_abstraction_map,
 physical_connectivity_definition,
 physical_connectivity_definition_relationship,
 physical_connectivity_element,
 physical_connectivity_make_from_relationship,
 physical_network,
 physical_node_requirement_to_implementing_component_allocation,

physical_unit_relationship,
pitch_class,
placement_group,
placement_group_area_assignment,
planned_characteristic,
planned_effectivity,
positional_boundary_and_profile_boundary_member_relationship,
positional_boundary_definition,
positional_boundary_member_definition,
positional_boundary_offset,
printed_connector_template_terminal_relationship,
printed_part_template_connected_terminals_definition,
printed_part_template_material,
printed_part_template_material_link,
printed_part_template_terminal,
printed_part_template_terminal_relationship,
product_association,
projected_zone_and_base_relationship,
projected_zone_height_characteristic,
promissory_usage_relationship,
promissory_usage_in_product_model_relationship,
reference_composition_path,
reference_functional_unit_assignment_to_part,
registered_font,
requirement_allocation,
requirement_composition,
restraint_condition,
rule_action,
rule_justification,
rule_priority,
rule_set,
runout_tolerance_zone_orienting_relationship,
runout_zone_orientation,
security_classification,
scalar_terminal_definition_link,
separate_geometric_tolerance_requirement,
shape_definition,
shape_dimension_representation,
shape_element,
shape_element_locating_relationship,
shape_element_relationship,
shape_item,
signal,
signal_property,
signal_specification,
simultaneous_geometric_tolerance_requirement,
solid_curve,
solid_fill_style,
specification_allocation,
stratum_concept_relationship,
stratum_concept_to_physical_usage_view_assignment,
stratum_feature_in_stratum_3d_position,

```

stratum_feature_in_stratum_planar_position,
stratum_feature_shape_stratum_average_surface_shape_relationship,
stratum_feature_shape_stratum_surface_shape_relationship,
stratum_feature_styled_area_representation,
stratum_interconnect_module_3d_position,
stratum_interconnect_module_planar_position,
stratum_make_from_relationship,
stratum_average_surface,
stratum_surface,
stratum_surface_in_stratum_3d_position,
stratum_technology,
stratum_technology_link,
structured_text,
styled_curve_occurrence,
substitute_product,
sub_assembly_relationship,
supplied_product_version,
technology,
template_stratum_assignment,
terminal_pair_port_equivalence_relationship,
termination_constraint,
thermal_isolation_requirement_assignment,
tolerance_zone_and_viewing_plane_orienting_relationship,
tolerance_zone_boundary_definition,
tolerance_zone_boundary_member_offset,
tolerance_zone_definition,
tolerance_zone_form,
tolerance_zone_per_unit_size_specification,
topological_junction,
transmission_line_functional_unit_terminal_allocation,
true_profile,
usage_concept_usage_relationship,
usage_view_connection_zone_terminal_shape_relationship,
vector,
viewing_plane_based_orientation,
viewing_plane_definition,
volume_shape_intersection,
volume_shape_plane_projection,
volume_shape_point_projection,
work_order,
work_request) ANDOR
frozen_design_object) ANDOR
owned_design_object);
END_ENTITY; -- managed_design_object

```

```

ENTITY manifold_subsurface_shape
  SUBTYPE OF (shape_definition);
END_ENTITY; -- manifold_subsurface_shape

```

```

ENTITY marking

```

```

SUBTYPE OF (managed_design_object);
    marked_message      : marked_message_select;
    marked_device_shape : view_shape_select;
    marking_material    : ee_material;
    placement           : axis_placement;
    message_representation : feature_shape_definition;
    message_representation_text : styled_text;
    marking_specification : ee_specification;
WHERE
    WR1 : SIZEOF(QUERY(st <* message_representation\shape_definition.element | st = message_representation_text)) = 1;
    WR2 : SIZEOF(QUERY(ap <* marked_device_shape\shape_definition.element | ap = placement)) = 1;
    WR3 : SIZEOF(TYPEOF(marked_device_shape) * ['AP210_ARM.PHYSICAL_UNIT_3D_SHAPE',
'AP210_ARM.PHYSICAL_UNIT_PLANAR_SHAPE']) = 1;
END_ENTITY; -- marking

```

```

ENTITY mass_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- mass_data_element

```

```

ENTITY material_addition_feature
    SUBTYPE OF (design_intent_modification);
    material_addition_shape : material_addition_feature_template;
END_ENTITY; -- material_addition_feature

```

```

ENTITY material_addition_feature_template
    SUBTYPE OF (part_template);
    of_stratum_technology : stratum_technology;
END_ENTITY; -- material_addition_feature_template

```

```

ENTITY material_composition_relationship
    SUBTYPE OF (managed_design_object);
    composition_basis      : STRING;
    determination_method : ee_text;
    composition_class      : STRING;
    constituent_amount     : data_element;
    related_product        : ee_material;
    relating_product       : ee_material;
END_ENTITY; -- material_composition_relationship

```

```

ENTITY material_definition
    SUBTYPE OF (ee_product_definition);
    material : ee_material;
END_ENTITY; -- material_definition

```

```

ENTITY material_electrical_conductivity_requirement_occurrence
  SUBTYPE OF (electrical_requirement_occurrence);
    electrical_conductivity_characteristic : material_conductivity_classification;
END_ENTITY; -- material_electrical_conductivity_requirement_occurrence

```

```

ENTITY material_removal_feature
  SUPERTYPE OF (filled_area_material_removal_structure)
  SUBTYPE OF (design_intent_modification);
    material_removal_shape : material_removal_feature_template;
END_ENTITY; -- material_removal_feature

```

```

ENTITY material_removal_feature_template
  SUPERTYPE OF (ONEOF (electrical_isolation_removal_template,
thermal_isolation_removal_template))
  SUBTYPE OF (part_template);
    of_stratum_technology : stratum_technology;
END_ENTITY; -- material_removal_feature_template

```

```

ENTITY material_removal_laminate_component
  SUPERTYPE OF (filled_area_material_removal_laminate_component)
  SUBTYPE OF (laminate_component);
    SELF\assembly_component.usage_definition : material_removal_feature_template;
    design_intent : stratum_feature;
END_ENTITY; -- material_removal_laminate_component

```

```

ENTITY material_shape_modification
  SUBTYPE OF (design_intent_modification);
END_ENTITY; -- material_shape_modification

```

```

ENTITY material_specification
  SUPERTYPE OF (surface_finish_specification)
  SUBTYPE OF (ee_specification);
END_ENTITY; -- material_specification

```

```

ENTITY mating_connector_component
  SUBTYPE OF (managed_design_object);
    definition : mating_connector_part;
    reference_designation : OPTIONAL ee_name;
  INVERSE
    placement_context : mating_connector_placement_relationship FOR placed_connector;
END_ENTITY; -- mating_connector_component

```

```

ENTITY mating_connector_placement_relationship
  SUBTYPE OF (managed_design_object);
    placed_connector      : mating_connector_component;
    interface_context     : next_higher_assembly_interface_requirement;
    connector_placement   : axis_placement;
END_ENTITY; -- mating_connector_placement_relationship

```

```

ENTITY mating_connector_termination
  SUBTYPE OF (managed_design_object);
    definition            : packaged_or_printed_terminal;
    accessed_connector    : mating_connector_component;
    name                  : ee_name;
    signal_definition     : OPTIONAL ee_requirement_occurrence;
  UNIQUE
    UR1 : definition,
accessed_connector;
END_ENTITY; -- mating_connector_termination

```

```

ENTITY maximum_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- maximum_dimension_representation

```

```

ENTITY maximum_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET[1:1] OF numeric_param
END_ENTITY; -- maximum_tolerance_characteristic

```

```

ENTITY measure_qualification
  SUBTYPE OF (managed_design_object);
    name                  : text;
    description           : text;
    qualified_measure     : data_element;
    qualifiers            : SET[1:?] OF value_qualifier;
END_ENTITY; -- measure_qualification

```

```

ENTITY mechanical_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- mechanical_requirement_occurrence

```

```

ENTITY minimally_defined_bare_die_terminal
  SUPERTYPE OF (guided_wave_bare_die_terminal ANDOR
bare_die_terminal)
  SUBTYPE OF (part_terminal);
    SELF\part_feature.associated_definition : bare_die;
END_ENTITY; -- minimally_defined_bare_die_terminal

```

```

ENTITY minimally_defined_component_terminal
  SUBTYPE OF (physical_component_terminal);
  SELF\component_feature.definition : part_terminal_external_reference;
END_ENTITY; -- minimally_defined_component_terminal

```

```

ENTITY minimum_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- minimum_dimension_representation

```

```

ENTITY minimum_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
  SELF\coordinated_characteristic.component_characteristic : SET[1:1] OF numeric_param
END_ENTITY; -- minimum_tolerance_characteristic

```

```

ENTITY model_parameter
  SUPERTYPE OF (datum_difference_based_model_parameter)
  SUBTYPE OF (managed_design_object);
  type_name          : STRING;
  unit_of_measure    : OPTIONAL STRING;
  reference_document : ee_document;
  valid_range        : OPTIONAL SET[1:?] OF coordinated_characteristic;
  default_value      : OPTIONAL scalar_or_coordinated_characteristics;
  UNIQUE
    UR1 : type_name,
reference_document;
END_ENTITY; -- model_parameter

```

```

ENTITY mounting_restriction_area
  SUBTYPE OF (non_feature_shape_element);
  mounting_surface : interconnect_module_component_surface_feature
  SELF\non_feature_shape_element.scope : assembly_module;
  WHERE
    WR1 : 'AP210_ARM.CSG_2D_SHAPE' IN TYPEOF(SELF\non_feature_shape_element.element_shape
END_ENTITY; -- mounting_restriction_area

```

```

ENTITY mounting_restriction_volume
  SUBTYPE OF (non_feature_shape_element);
  mounting_surface : interconnect_module_component_surface_feature
  SELF\non_feature_shape_element.scope : assembly_module;
  INVERSE
    volume : non_feature_shape_definition FOR associated_element;
  WHERE
    WR1 : 'AP210_ARM.BOUND_VOLUME_SHAPE' IN TYPEOF(volume);
END_ENTITY; -- mounting_restriction_volume

```



```

ENTITY multi_layer_material_removal_laminate_component
  SUBTYPE OF (material_removal_laminate_component);
    derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_material_removal_laminate_component

```

```

ENTITY multi_layer_stratum_feature
  SUBTYPE OF (stratum_feature);
    derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_stratum_feature

```

```

ENTITY multi_layer_stratum_feature_template_component
  SUBTYPE OF (stratum_feature_template_component);
    derived_layer_geometry : layer;
END_ENTITY; -- multi_layer_stratum_feature_template_component

```

```

ENTITY next_higher_assembly_interface_requirement
  SUBTYPE OF (managed_design_object);
    shape : bound_volume_shape;
    version_of_next_higher_assembly : ee_product_version;
END_ENTITY; -- next_higher_assembly_interface_requirement

```

```

ENTITY next_higher_assembly_relationship
  SUBTYPE OF (assembly_composition_relationship);
    reference_designation : OPTIONAL ee_name;
    quantity : OPTIONAL data_element;
  INVERSE
    assembly_position_3d : SET[0:1] OF component_assembly_3d_position FOR component;
    assembly_position_2d : SET[0:1] OF component_assembly_2d_position FOR component;
  UNIQUE
    UR1 : reference_designation,
assembly;
  WHERE
    WR1 : (EXISTS(reference_designation)) XOR (EXISTS(quantity));
END_ENTITY; -- next_higher_assembly_relationship

```

```

ENTITY non_conductive_base_blind_via
  SUBTYPE OF (blind_via);
END_ENTITY; -- non_conductive_base_blind_via

```

```

ENTITY non_conductive_cross_section_template
  SUBTYPE OF (part_template);
    template_technology : stratum_technology;
    nominal_width : length_data_element;
END_ENTITY; -- non_conductive_cross_section_template

```

```

ENTITY non_feature_shape_definition
  SUPERTYPE OF ((ONEOF (pre_defined_datum_axis_3d_symbol,
pre_defined_datum_plane_3d_symbol,
pre_defined_parallel_datum_axis_2d_symbol,
pre_defined_perpendicular_datum_axis_2d_symbol,
pre_defined_perpendicular_datum_plane_2d_symbol,
view_level_non_feature_shape_definition)) ANDOR
(ONEOF (physical_unit_3d_keepout_shape,
physical_unit_planar_keepout_shape)))
  SUBTYPE OF (shape_definition);
    associated_element : non_feature_shape_element;
    model_shape       : OPTIONAL feature_shape_definition;
END_ENTITY; -- non_feature_shape_definition

```

```

ENTITY non_feature_shape_element
  SUPERTYPE OF (ONEOF (datum_reference_frame,
datum_system,
derived_shape_element,
directed_axis,
positional_boundary,
positional_boundary_member,
tolerance_zone,
tolerance_zone_boundary,
tolerance_zone_boundary_member,
viewing_plane,
seating_plane,
template_material_cross_section_boundary,
interconnect_module_constraint_region,
mounting_restriction_area,
mounting_restriction_volume) ANDOR
datum)
  SUBTYPE OF (shape_element);
    scope : ee_product_definition_with_annotation_elements;
  INVERSE
    element_shape : SET[0:?] OF non_feature_shape_definition FOR associated_e
    associated_usage_concept : SET[0:?] OF usage_concept_usage_relationship FOR associat
  UNIQUE
    UR1 : SELF\shape_element.name,
scope;
  WHERE
    WR1 : (NOT (SIZEOF(associated_usage_concept) > 0)) OR (SIZEOF(element_shape) > 0);
END_ENTITY; -- non_feature_shape_element

```

```

ENTITY non_feature_shape_element_relationship
  SUBTYPE OF (shape_element_relationship);
    SELF\shape_element_relationship.related_shape_element : non_feature_shape_element;
    SELF\shape_element_relationship.relatng_shape_element : non_feature_shape_element;
  UNIQUE
    UR1 : related_shape_element,

```

```
relating_shape_element;
END_ENTITY; -- non_feature_shape_element_relationship
```

```
ENTITY non_functional_land
  SUPERTYPE OF (ONEOF (via_dependent_non_functional_land,
    component_termination_passage_dependent_non_functional_land,
    contact_size_dependent_non_functional_land,
    component_termination_passage_and_contact_size_dependent_non_functional_land,
    unsupported_passage_dependent_non_functional_land,
    via_and_contact_size_dependent_non_functional_land))
  SUBTYPE OF (stratum_feature_template_component);
    alternate_land_definition : OPTIONAL land_physical_template;
    SELF\assembly_component.usage_definition : land_physical_template;
  INVERSE
    access_mechanism : SET[1:?] OF non_functional_land_join_terminal FOR associated_component;
END_ENTITY; -- non_functional_land
```

```
ENTITY non_functional_land_interface_terminal
  SUBTYPE OF (laminate_component_interface_terminal);
    SELF\component_feature.definition : land_template_interface_terminal;
    SELF\component_feature.associated_component : non_functional_land;
  WHERE
    WR1 : SIZEOF(['AP210_ARM.CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND',
    'AP210_ARM.COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND',
    'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_LAND'] * TYPEOF(SELF\component_feature.definition)) > 0;
END_ENTITY; -- non_functional_land_interface_terminal
```

```
ENTITY non_functional_land_join_terminal
  SUBTYPE OF (laminate_component_join_terminal);
    SELF\component_feature.definition : land_template_join_terminal;
    SELF\component_feature.associated_component : non_functional_land;
END_ENTITY; -- non_functional_land_join_terminal
```

```
ENTITY numeric_parameter
  SUBTYPE OF (characteristic);
    SELF\characteristic.property_value : data_element;
END_ENTITY; -- numeric_parameter
```

```
ENTITY open_shell_based_surface
  SUPERTYPE OF (ONEOF (stratum_average_surface_shape,
    stratum_surface_shape,
    design_intent_modification_non_planar_2d_shape,
    part_template_non_planar_2d_shape,
    stratum_feature_non_planar_2d_shape))
  SUBTYPE OF (shape_definition);
END_ENTITY; -- open_shell_based_surface
```

```

ENTITY operational_requirement_occurrence_relationship
  SUBTYPE OF (managed_design_object);
    relationship_type          : relationship_operation;
    relating_requirement_occurrence : ee_requirement_occurrence;
    related_requirement_occurrence  : ee_requirement_occurrence;
END_ENTITY; -- operational_requirement_occurrence_relationship

```

```

ENTITY opposing_boundary_set_size_characteristic
  SUBTYPE OF (boundary_size_characteristic);
    sized_boundary_set : tolerance_zone_opposing_boundary_set;
END_ENTITY; -- opposing_boundary_set_size_characteristic

```

```

ENTITY organization
  SUBTYPE OF (managed_design_object);
    identifier : ee_name;
    address    : OPTIONAL STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- organization

```

```

ENTITY owned_design_object
  SUBTYPE OF (managed_design_object);
  INVERSE
    owner : owner_assignment FOR owned_item;
END_ENTITY; -- owned_design_object

```

```

ENTITY owner_assignment
  SUBTYPE OF (managed_design_object);
    owned_item      : SET[1:?] OF owned_design_object;
    assigned_owner  : person_and_organization;
END_ENTITY; -- owner_assignment

```

```

ENTITY movable_packaged_component_join_terminal
  SUBTYPE OF (packaged_component_join_terminal);
    maximum_wire_terminal_length : length_data_element;
    minimum_wire_terminal_length : length_data_element;
  WHERE
    WR1 : 'AP210_ARM.WIRE_TERMINAL' IN TYPEOF(SELF\component_feature.definition\packaged_
END_ENTITY; -- movable_packaged_component_join_terminal

```

```

ENTITY package
  SUPERTYPE OF (altered_package)
  SUBTYPE OF (physical_unit_usage_view);

```

```

mounting_technology                : mounting_technology_type;
maximum_seating_plane_installation_offset : OPTIONAL length_data_element;
surface_mount_lead_form            : OPTIONAL lead_form_type;
nominal_mounting_lead_pitch        : OPTIONAL length_data_element;
nominal_mounting_lead_span         : OPTIONAL length_data_element;
least_material_condition_centroid_location : OPTIONAL cartesian_point;
maximum_material_condition_centroid_location : OPTIONAL cartesian_point;
maximum_body_height_above_seating_plane : length_data_element;
maximum_body_height_below_seating_plane : OPTIONAL length_data_element;
maximum_body_clearance_above_seating_plane : OPTIONAL length_data_element;
maximum_body_clearance_below_seating_plane : OPTIONAL length_data_element;
minimum_body_clearance_above_seating_plane : OPTIONAL length_data_element;
minimum_body_clearance_below_seating_plane : OPTIONAL length_data_element;
maximum_lead_length_below_seating_plane : OPTIONAL length_data_element;
least_lead_length_below_seating_plane : OPTIONAL length_data_element;
of_datum_reference_frame           : OPTIONAL datum_reference_frame;
of_defining_datum_system_relationship : OPTIONAL SET[1:3] OF datum_feature;
of_derived_datum_system_defining_relationship : OPTIONAL SET[1:3] OF datum_usage_in

DERIVE
    maximum_package_total_vertical_extent : length_data_element := max_data_element(add_
maximum_body_height_below_seating_plane),
add_data_element(maximum_body_height_above_seating_plane,
maximum_lead_length_below_seating_plane));
    cutout_required                : BOOLEAN := (maximum_body_height_below_seating
    maximum_installed_height        : length_data_element := add_data_element(maxim
maximum_seating_plane_installation_offset);
INVERSE
    body                            : SET[0:1] OF package_body FOR associated_defini
    package_seating_plane           : non_feature_shape_element FOR scope;
    interface_plane                 : SET[0:1] OF non_feature_shape_element FOR sc
    package_accesses                : SET[0:?] OF package_terminal FOR associated_
    package_polarity_indication     : SET[0:1] OF part_feature FOR associated_defini
    package_primary_orientation_feature : SET[0:1] OF part_feature FOR associated_defini
    package_secondary_orientation_feature : SET[0:1] OF part_feature FOR associated_defini
    package_tertiary_orientation_feature : SET[0:1] OF part_feature FOR associated_defini
WHERE
    WR1 : NOT (EXISTS(package_primary_orientation_feature)) OR ('AP210_ARM.PRIMARY_ORIE
    WR2 : NOT (EXISTS(package_secondary_orientation_feature)) OR ('AP210_ARM.SECONDARY_
    WR3 : NOT (EXISTS(package_tertiary_orientation_feature)) OR ('AP210_ARM.TERTIARY_OR
    WR4 : 'AP210_ARM.SEATING_PLANE' IN TYPEOF(package_seating_plane);
    WR5 : 'AP210_ARM.SEATING_PLANE' IN TYPEOF(interface_plane);
    WR6 : NOT (mounting_technology = surface_mount) OR (NOT (EXISTS(maximum_lead_length
    WR7 : NOT (EXISTS(maximum_lead_length_below_seating_plane)) OR (maximum_lead_length
    WR8 : NOT (EXISTS(least_lead_length_below_seating_plane)) OR (least_lead_length_bel
    WR9 : (NOT (mounting_technology = surface_mount) AND NOT (EXISTS(surface_mount_lead
    WR10 : package_seating_plane\shape_element.name = 'package seating plane';
    WR11 : NOT (EXISTS(interface_plane[1])) OR (interface_plane\shape_element.name = 'int
    WR12 : NOT (EXISTS(interface_plane[1])) OR (package_seating_plane <> interface_plane
END_ENTITY; -- package

```

ENTITY package_body

```

    SUBTYPE OF (part_feature);
        body_material                : ee_material;
        SELF\part_feature.associated_definition : package;
    WHERE
        WR1 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') | ('AP210_ARM.' + 'PACKAGE_BODY
        WR2 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') | ('AP210_ARM.' + 'PACKAGE_BODY
        WR3 : SIZEOF(QUERY(pf <* USEDIN(SELF,
'AP210_ARM.PACKAGE_BODY_SURFACE.ASSOCIATED_PACKAGE_BODY') | ('AP210_ARM.' + 'PACKAGE_BODY
    END_ENTITY; -- package_body

```

```

ENTITY package_body_bottom_surface
    SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_bottom_surface

```

```

ENTITY package_body_edge_segment_surface
    SUBTYPE OF (package_body_surface);
        composed_surface : package_body_edge_surface;
        start_vertex      : edge_segment_vertex;
        end_vertex        : edge_segment_vertex;
    WHERE
        WR1 : start_vertex :<>: end_vertex;
END_ENTITY; -- package_body_edge_segment_surface

```

```

ENTITY package_body_edge_surface
    SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_edge_surface

```

```

ENTITY package_body_surface
    ABSTRACT SUPERTYPE OF (ONEOF (package_body_bottom_surface,
package_body_top_surface,
package_body_edge_segment_surface,
package_body_edge_surface))
    SUBTYPE OF (part_feature);
        associated_package_body : package_body;
END_ENTITY; -- package_body_surface

```

```

ENTITY package_body_top_surface
    SUBTYPE OF (package_body_surface);
END_ENTITY; -- package_body_top_surface

```

```

ENTITY package_external_reference
    SUBTYPE OF (managed_design_object);

```

```

    design_owner          : STRING;
    part_number           : STRING;
    revision_code         : STRING;
    product_definition_id : STRING;
END_ENTITY; -- package_external_reference

```

```

ENTITY package_terminal
  SUPERTYPE OF (((altered_package_terminal ANDOR
guided_wave_terminal) ANDOR
wire_terminal) ANDOR
primary_reference_terminal)
  SUBTYPE OF (part_feature);
    SELF\part_feature.associated_definition : package;
    external_connection_zone               : OPTIONAL SET[1:?] OF connection_zone;
    internal_connection_zone               : OPTIONAL SET[1:?] OF connection_zone;
    seating_plane_intersection             : seating_plane_intersection_type;
    terminal_surface_material              : material_definition;
    model_representation                   : OPTIONAL SET[1:?] OF analytical_representation;
    terminal_characteristic                 : OPTIONAL SET[1:?] OF characteristic;
    terminal_core_material                  : OPTIONAL material_definition;
    minimum_terminal_diametrical_extent    : OPTIONAL length_data_element;
    maximum_terminal_diametrical_extent    : OPTIONAL length_data_element;
    seating_plane_zone                     : OPTIONAL connection_zone_interface_plane_representation;
  DERIVE
    mates_with_substrate : BOOLEAN := (EXISTS(seating_plane_zone)) AND (associated_definition = associated_definition);
  INVERSE
    SELF\part_feature.feature_shape : SET[1:?] OF usage_concept_usage_relationship FOR associated_definition;
  WHERE
    WR1 : NOT (EXISTS(seating_plane_zone)) OR (SIZEOF(QUERY(cz <* external_connection_zone)) = 0);
    WR2 : NOT (EXISTS(seating_plane_zone)) OR (SIZEOF(QUERY(cz <* internal_connection_zone)) = 0);
    WR3 : NOT (EXISTS(seating_plane_zone)) OR ((associated_definition.package_seating_plane_intersection = seating_plane_intersection_type));
    WR4 : NOT (EXISTS(seating_plane_zone)) OR ((seating_plane_intersection = surface_intersection_type));
    WR5 : NOT ((seating_plane_intersection = does_not_intersect) AND (EXISTS(seating_plane_zone)));
END_ENTITY; -- package_terminal

```

```

ENTITY package_terminal_external_reference
  SUBTYPE OF (managed_design_object);
    associated_definition : package_external_reference;
    name                  : STRING;
  UNIQUE
    UR1 : associated_definition,
name;
END_ENTITY; -- package_terminal_external_reference

```

```

ENTITY package_terminal_occurrence
  SUBTYPE OF (managed_design_object);
    definition          : package_terminal;
    associated_component : packaged_component;
END_ENTITY; -- package_terminal_occurrence

```

```

ENTITY package_terminal_surface_constituent_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relate_shape_element : package_body_surface;
  SELF\shape_element_relationship.related_shape_element : package_terminal;
END_ENTITY; -- package_terminal_surface_constituent_relationship

```

```

ENTITY packaged_component
  SUPERTYPE OF (packaged_connector_component)
  SUBTYPE OF (physical_component);
  SELF\assembly_component.usage_definition : packaged_part;
  selected_package_alternate : OPTIONAL altered_package;
  INVERSE
    access_mechanism : SET[1:?] OF packaged_component_join_terminal FOR associated_component;
END_ENTITY; -- packaged_component

```

```

ENTITY packaged_component_join_terminal
  SUPERTYPE OF (movable_packaged_component_join_terminal)
  SUBTYPE OF (physical_component_terminal);
  SELF\component_feature.definition : packaged_part_join_terminal;
  SELF\component_feature.associated_component : packaged_component;
END_ENTITY; -- packaged_component_join_terminal

```

```

ENTITY packaged_connector
  SUBTYPE OF (packaged_part);
  INVERSE
    characterized_connector_terminal_relationship : SET[1:?] OF packaged_connector_terminal;
END_ENTITY; -- packaged_connector

```

```

ENTITY packaged_connector_component
  SUBTYPE OF (interface_component,
packaged_component);
  SELF\assembly_component.usage_definition : packaged_connector;
  INVERSE
    interface_access_mechanism : SET[1:?] OF packaged_connector_component_interface_terminal;
END_ENTITY; -- packaged_connector_component

```

```

ENTITY packaged_connector_component_interface_terminal
  SUBTYPE OF (component_interface_terminal);
  SELF\component_feature.definition : packaged_part_interface_terminal;
  SELF\component_feature.associated_component : packaged_connector_component;
END_ENTITY; -- packaged_connector_component_interface_terminal

```

```

ENTITY packaged_connector_terminal_relationship

```



```

SUBTYPE OF (managed_design_object);
    connector          : packaged_connector;
    join_terminal      : packaged_part_join_terminal;
    interface_terminal : packaged_part_interface_terminal;
UNIQUE
    UR1 : join_terminal,
connector;
    UR2 : interface_terminal,
connector;
END_ENTITY; -- packaged_connector_terminal_relationship

```

```

ENTITY packaged_part
    SUPERTYPE OF (packaged_connector ANDOR
altered_packaged_part)
    SUBTYPE OF (physical_unit_usage_view);
        used_package          : SET[1:?] OF package_or_package_external_reference;
        potting_compound      : OPTIONAL ee_material;
        implemented_function   : OPTIONAL functional_unit_usage_view;
    INVERSE
        access_mechanisms : SET[0:?] OF packaged_part_terminal FOR associated_definition;
    WHERE
        WR1 : NOT (EXISTS(implemented_function) XOR (SIZEOF(access_mechanisms) > 0));
        WR2 : single_package_version(SELF);
END_ENTITY; -- packaged_part

```

```

ENTITY packaged_part_interface_terminal
    SUBTYPE OF (packaged_part_terminal);
    INVERSE
        external_connector_function : SET[1:?] OF packaged_connector_terminal_relationship FOR
END_ENTITY; -- packaged_part_interface_terminal

```

```

ENTITY packaged_part_join_terminal
    SUBTYPE OF (packaged_part_terminal);
    INVERSE
        internal_connector_function : SET[0:1] OF packaged_connector_terminal_relationship FOR
END_ENTITY; -- packaged_part_join_terminal

```

```

ENTITY packaged_part_terminal
    ABSTRACT SUPERTYPE OF (ONEOF (packaged_part_interface_terminal,
packaged_part_join_terminal))
    SUBTYPE OF (part_terminal);
        terminal_of_package          : SET[1:?] OF package_terminal_or_package_terminal;
        SELF\part_feature.associated_definition : packaged_part;
    WHERE
        WR1 : disjoint_package_terminals(SELF);
        WR2 : NOT EXISTS(SELF\part_feature.precedent_feature);
END_ENTITY; -- packaged_part_terminal

```

```

ENTITY padstack_definition
  SUPERTYPE OF (ONEOF (design_layer_type_specific_padstack_definition,
stratum_type_independent_padstack_definition,
stratum_occurrence_specific_padstack_definition))
  SUBTYPE OF (ee_product_definition);
    reference_implementation      : reference_packaged_part_interconnect_implementation;
    teardrop_taper_start_distance : OPTIONAL length_data_element;
    teardrop_taper_end_distance   : OPTIONAL length_data_element;
    snowball_start_distance       : OPTIONAL length_data_element;
    snowball_end_distance         : OPTIONAL length_data_element;
    snowball_initial_spacing      : OPTIONAL length_data_element;
    snowball_end_spacing          : OPTIONAL length_data_element;
    snowball_initial_radius       : OPTIONAL length_data_element;
    snowball_end_radius           : OPTIONAL length_data_element;
    snowball_quantity_per_trace   : OPTIONAL positive_integer;
  INVERSE
    shape : padstack_definition_shape FOR shape_characterized_padstack_definition;
  WHERE
    WR1 : SIZEOF(QUERY(prpc <* USEDIN(SELf\ee_product_definition.version.versioned_item,
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | (prpc.name = 'template mode
END_ENTITY; -- padstack_definition

```

```

ENTITY padstack_definition_shape
  SUBTYPE OF (planar_projected_shape);
    shape_characterized_padstack_definition : padstack_definition;
    shape_environment                       : OPTIONAL environment;
    shape_material_condition                 : material_condition;
END_ENTITY; -- padstack_definition_shape

```

```

ENTITY padstack_location_in_footprint_definition
  SUBTYPE OF (managed_design_object);
    padstack      : padstack_definition;
    footprint      : footprint_definition;
    reference_package_terminal : package_terminal;
END_ENTITY; -- padstack_location_in_footprint_definition

```

```

ENTITY padstack_location_in_footprint_definition_transform
  SUBTYPE OF (managed_design_object);
    padstack_shape      : padstack_definition_shape;
    footprint_shape      : footprint_definition_shape;
    reference_location   : padstack_location_in_footprint_definition;
    transform             : axis_placement_2d;
END_ENTITY; -- padstack_location_in_footprint_definition_transform

```

```

ENTITY padstack_occurrence
  SUBTYPE OF (assembly_group_component);

```

```

        SELF\assembly_component.usage_definition : padstack_definition;
WHERE
    WR1 : SIZEOF(TYPEOF(SELF) * ['AP210_ARM.PRINTED_COMPONENT',
'AP210_ARM.LAMINATE_COMPONENT',
'AP210_ARM.PHYSICAL_COMPONENT']) = 0;
END_ENTITY; -- padstack_occurrence

```

```

ENTITY padstack_occurrence_sub_assembly_relationship
    SUBTYPE OF (managed_design_object);
    related_component          : laminate_component;
    relating_component         : padstack_occurrence;
    sub_assembly_reference_designation : STRING;
    UNIQUE
        UR1 : sub_assembly_reference_designation,
relating_component,
related_component;
END_ENTITY; -- padstack_occurrence_sub_assembly_relationship

```

```

ENTITY parallel_shape_element_relationship
    SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- parallel_shape_element_relationship

```

```

ENTITY parallelism_tolerance
    SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- parallelism_tolerance

```

```

ENTITY parameter_assignment
    SUBTYPE OF (managed_design_object);
    parameter_value      : scalar_or_coordinated_characteristics;
    assigned_parameter : model_parameter;
END_ENTITY; -- parameter_assignment

```

```

ENTITY part_array_feature
    ABSTRACT SUPERTYPE OF (ONEOF (part_linear_array_feature,
part_rectangular_array_feature))
    SUBTYPE OF (part_group_feature);
END_ENTITY; -- part_array_feature

```

```

ENTITY part_connected_terminals_definition
    SUBTYPE OF (managed_design_object);
    associated_definition : physical_unit_usage_view;
    connected_terminals  : SET[2:?] OF part_terminal;
    name                  : ee_name;
    UNIQUE

```

```

        UR1 : name,
associated_definition;
WHERE
    WR1 : SIZEOF(QUERY(ct <* connected_terminals | NOT (associated_definition = ct\part_
END_ENTITY; -- part_connected_terminals_definition

```

```

ENTITY part_device
    SUBTYPE OF (managed_design_object);
    defined_packaged_part : packaged_part;
    included_device       : bare_die;
    item_identification   : ee_name;
    bonding_agent         : OPTIONAL ee_material;
    INVERSE
        device_3d_position : SET[0:1] OF device_3d_position FOR placed_device;
        device_2d_position : SET[0:1] OF device_2d_position FOR placed_device;
    UNIQUE
        UR1 : defined_packaged_part,
item_identification;
END_ENTITY; -- part_device

```

```

ENTITY part_device_terminal
    SUBTYPE OF (managed_design_object);
    used_bare_die_terminal : minimally_defined_bare_die_terminal;
    terminated_device      : part_device;
    UNIQUE
        UR1 : used_bare_die_terminal,
terminated_device;
END_ENTITY; -- part_device_terminal

```

```

ENTITY part_device_terminal_model_port_assignment
    SUBTYPE OF (managed_design_object);
    internal_connection : part_device_terminal;
    internal_port        : analytical_model_port;
    UNIQUE
        UR1 : internal_connection,
internal_port;
END_ENTITY; -- part_device_terminal_model_port_assignment

```

```

ENTITY part_feature
    SUPERTYPE OF ((((((((((part_group_feature ANDOR
ONEOF (primary_orientation_feature,
secondary_orientation_feature,
tertiary_orientation_feature,
package_body)) ANDOR
part_tooling_feature) ANDOR
thermal_feature) ANDOR
part_interface_access_feature) ANDOR

```

```

part_mating_feature) ANDOR
part_mounting_feature) ANDOR
part_terminal) ANDOR
interconnect_module_surface_feature) ANDOR
package_terminal) ANDOR
package_body_surface) ANDOR
polarity_indication_feature)
    SUBTYPE OF (physical_feature_or_part_template);
        associated_definition      : physical_unit_usage_view;
        material_state_change      : OPTIONAL material_state_change_enumeration;
        precedent_feature           : OPTIONAL part_feature;
        SELF\shape_element.name    : ee_name;
    INVERSE
        design_view_definition_stratum_concept      : SET[0:1] OF stratum_concept_to_physical_u
        design_view_definition_component_feature    : SET[0:1] OF component_feature_to_physical
        feature_shape                               : SET[0:?] OF usage_concept_usage_relations
    UNIQUE
        UR1 : SELF\shape_element.name,
associated_definition;
END_ENTITY; -- part_feature

```

```

ENTITY part_feature_make_from_relationship
    SUBTYPE OF (managed_design_object);
        associated_make_from : engineering_make_from;
        reusable_feature      : part_feature;
        resultant_feature     : part_feature;
    UNIQUE
        UR1 : reusable_feature,
resultant_feature;
    WHERE
        WR1 : reusable_feature :<>: resultant_feature;
        WR2 : reusable_feature.associated_definition :<>: resultant_feature.associated_defin
END_ENTITY; -- part_feature_make_from_relationship

```

```

ENTITY part_feature_model_port_assignment
    SUBTYPE OF (managed_design_object);
        source : part_feature;
        target : analytical_model_port;
END_ENTITY; -- part_feature_model_port_assignment

```

```

ENTITY part_feature_placement_link
    SUBTYPE OF (managed_design_object);
        precedent_element : part_feature;
        subsequent_element : part_feature;
    INVERSE
        referenced_by : part_linear_array_feature FOR element;
END_ENTITY; -- part_feature_placement_link

```

```

ENTITY part_group_feature
  SUPERTYPE OF (part_array_feature)
  SUBTYPE OF (part_feature);
  WHERE
    WR1 : SIZEOF(QUERY(fgr <* USEDIN(SELF,
'AP210_ARM.FEATURE_GROUP_RELATIONSHIP.RELATING_FEATURE') | fgr.related_feature\part_featu
END_ENTITY; -- part_group_feature

```

```

ENTITY part_interface_access_feature
  SUBTYPE OF (part_feature);
  accessed_interface : part_feature;
  interface_access_area : OPTIONAL SET[1:?] OF connection_zone;
  WHERE
    WR1 : SELF\part_feature.material_state_change = material_removal;
    WR2 : SELF <> accessed_interface;
    WR3 : NOT ('AP210_ARM.PART_TERMINAL' IN TYPEOF(SELF));
    WR4 : NOT ('AP210_ARM.PACKAGE_BODY' IN TYPEOF(SELF));
    WR5 : NOT ('AP210_ARM.PART_MOUNTING_FEATURE' IN TYPEOF(SELF));
    WR6 : NOT ('AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE' IN TYPEOF(SELF));
    WR7 : NOT ('AP210_ARM.PACKAGE_TERMINAL' IN TYPEOF(SELF));
    WR8 : NOT ('AP210_ARM.PACKAGE_BODY_SURFACE' IN TYPEOF(SELF));
    WR9 : NOT ('AP210_ARM.POLARITY_INDICATION_FEATURE' IN TYPEOF(SELF));
END_ENTITY; -- part_interface_access_feature

```

```

ENTITY part_linear_array_feature
  SUBTYPE OF (part_array_feature);
  element : SET[1:?] OF part_feature_placement_link;
  initial_node_positive_x_displacement : length_data_element;
  terminus_node_positive_x_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- part_linear_array_feature

```

```

ENTITY part_linear_array_feature_link
  SUBTYPE OF (managed_design_object);
  precedent_element : part_linear_array_feature;
  subsequent_element : part_linear_array_feature;
  INVERSE
    referenced_by : part_rectangular_array_feature FOR element;
END_ENTITY; -- part_linear_array_feature_link

```

```

ENTITY part_mating_feature
  SUBTYPE OF (part_feature);
  mating_area : OPTIONAL SET[1:?] OF connection_zone;
  WHERE
    WR1 : NOT ('AP210_ARM.PART_MOUNTING_FEATURE' IN TYPEOF(SELF));
    WR2 : NOT ('AP210_ARM.INTERCONNECT_MODULE_SURFACE_FEATURE' IN TYPEOF(SELF));

```

```

        WR3 : NOT ('AP210_ARM.PACKAGE_TERMINAL' IN TYPEOF(SELF));
        WR4 : NOT ('AP210_ARM.PART_TERMINAL' IN TYPEOF(SELF));
END_ENTITY; -- part_mating_feature

```

```

ENTITY part_mounting_feature
  SUPERTYPE OF (bare_die_surface)
  SUBTYPE OF (part_feature);
  mounting_area : OPTIONAL SET[1:?] OF connection_zone;
END_ENTITY; -- part_mounting_feature

```

```

ENTITY part_rectangular_array_feature
  SUBTYPE OF (part_array_feature);
  element : SET[1:?] OF part_linear_array_feature_link;
  initial_row_positive_y_displacement : length_data_element;
  terminus_row_positive_y_displacement : length_data_element;
  location_uncertainty : length_data_element;
END_ENTITY; -- part_rectangular_array_feature

```

```

ENTITY part_template
  SUPERTYPE OF (ONEOF (inter_stratum_feature_template,
material_addition_feature_template,
material_removal_feature_template,
non_conductive_cross_section_template,
printed_part_cross_section_template,
printed_part_template,
special_symbol_part_template,
stratum_feature_template,
trace_template) ANDOR
part_text_template)
  SUBTYPE OF (physical_feature_or_part_template);
  part_template_analytical_model : OPTIONAL analytical_model;
  physical_characteristic : OPTIONAL SET[1:?] OF correlated_or_independent;
  analytical_part_template_representation : OPTIONAL SET[1:?] OF analytical_representation;
  associated_definition : template_definition;
  WHERE
    WR1 : SIZEOF(QUERY(prpc <* USEDIN(associated_definition\ee_product_definition.version
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | (prpc.name = 'template model
    WR2 : SELF\shape_element.name = 'NULL';
END_ENTITY; -- part_template

```

```

ENTITY part_template_3d_shape
  SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
  shape_material_condition : material_condition;
  shape_environment : OPTIONAL environment;
  shape_characterized_part_template : part_template;
END_ENTITY; -- part_template_3d_shape

```

```

ENTITY part_template_3d_keepout_shape
  SUBTYPE OF (bound_volume_shape,
non_feature_shape_definition);
  application_technology_constraint : OPTIONAL ee_requirement_occurrence;
  constrained_design_object_category : keepout_product_design_object_category;
  shape_characterized_part_template : part_template;
  shape_material_condition          : material_condition;
  shape_purpose                       : keepout_purpose;
  shape_environment                 : environment;
  centroid_location                 : OPTIONAL cartesian_point;
END_ENTITY; -- part_template_3d_keepout_shape

```

```

ENTITY part_template_location_in_padstack_definition
  SUBTYPE OF (managed_design_object);
  padstack          : padstack_definition;
  template          : template_definition;
  reference_designation : STRING;
END_ENTITY; -- part_template_location_in_padstack_definition

```

```

ENTITY part_template_location_in_padstack_definition_transform
  SUBTYPE OF (managed_design_object);
  padstack_shape      : padstack_definition_shape;
  template_shape      : part_template_planar_shape;
  reference_location   : part_template_location_in_padstack_definition;
  transform            : axis_placement_2d;
END_ENTITY; -- part_template_location_in_padstack_definition_transform

```

```

ENTITY part_template_non_planar_2d_shape
  SUBTYPE OF (open_shell_based_surface,
feature_shape_definition);
  transformation          : axis_placement_3d;
  shape_characterized_part_template : part_template;
  shape_environment       : OPTIONAL environment;
  shape_material_condition : material_condition;
END_ENTITY; -- part_template_non_planar_2d_shape

```

```

ENTITY part_template_or_physical_unit_2d_shape_representation_relationship
  SUBTYPE OF (managed_design_object);
  principal_part_shape : part_template_or_physical_unit_2d_shape_select;
  related_part_shape   : part_template_or_physical_unit_2d_shape_select;
  UNIQUE
  UR1 : principal_part_shape,
related_part_shape;
  WHERE
  WR1 : principal_part_shape :<>: related_part_shape;
END_ENTITY; -- part_template_or_physical_unit_2d_shape_representation_relationship

```



```

ENTITY part_template_or_physical_unit_3d_shape_representation_relationship
  SUBTYPE OF (managed_design_object);
    principal_part_shape : part_template_or_physical_unit_3d_shape_select;
    related_part_shape   : part_template_or_physical_unit_3d_shape_select;
  UNIQUE
    UR1 : principal_part_shape,
related_part_shape;
  WHERE
    WR1 : principal_part_shape :<>: related_part_shape;
END_ENTITY; -- part_template_or_physical_unit_3d_shape_representation_relationship

```

```

ENTITY part_template_planar_shape
  SUBTYPE OF (planar_projected_shape);
    shape_characterized_part_template : part_template;
    shape_environment                  : OPTIONAL environment;
    shape_material_condition           : material_condition;
END_ENTITY; -- part_template_planar_shape

```

```

ENTITY part_template_planar_keepout_shape
  SUBTYPE OF (planar_shape,
non_feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    constrained_design_object_category : keepout_product_design_object_category;
    shape_characterized_part_template : part_template;
    shape_material_condition           : material_condition;
    centroid_location                  : OPTIONAL cartesian_point;
    shape_environment                  : environment;
    shape_purpose                        : keepout_purpose;
END_ENTITY; -- part_template_planar_keepout_shape

```

```

ENTITY part_terminal
  ABSTRACT SUPERTYPE OF (ONEOF (minimally_defined_bare_die_terminal,
assembly_module_terminal,
interconnect_module_terminal,
packaged_part_terminal))
  SUBTYPE OF (part_feature);
  INVERSE
    connection_requirement : SET[0:1] OF part_connected_terminals_definition FOR
    functional_usage_view_terminal : SET[0:1] OF functional_usage_view_to_part_terminal_
END_ENTITY; -- part_terminal

```

```

ENTITY part_terminal_external_reference
  SUBTYPE OF (managed_design_object);
    associated_definition : physical_unit_or_part_template;
    feature_name           : ee_name;
    feature_type           : ee_name;

```

```

    UNIQUE
        UR1 : associated_definition,
feature_name,
feature_type;
END_ENTITY; -- part_terminal_external_reference

```

```

ENTITY part_text_template
    SUBTYPE OF (part_template);
        font : font_select;
        maximum_font_horizontal_extent : length_data_element;
        maximum_font_vertical_extent : length_data_element;
        slant_angle : angle_data_element;
        stroke_width : length_data_element;
END_ENTITY; -- part_text_template

```

```

ENTITY part_tooling_feature
    SUPERTYPE OF (fiducial_part_feature ANDOR
test_point_part_feature)
    SUBTYPE OF (part_feature);
END_ENTITY; -- part_tooling_feature

```

```

ENTITY partially_plated_cutout
    SUBTYPE OF (cutout);
    INVERSE
        plated_segment : SET[1:?] OF cutout_edge_segment FOR composed_cutout;
        unplated_segment : SET[1:?] OF cutout_edge_segment FOR composed_cutout;
END_ENTITY; -- partially_plated_cutout

```

```

ENTITY partially_plated_interconnect_module_edge
    SUBTYPE OF (interconnect_module_edge);
    INVERSE
        plated_segment : SET[1:?] OF interconnect_module_edge_segment FOR composed_edge;
        unplated_segment : SET[1:?] OF interconnect_module_edge_segment FOR composed_edge;
END_ENTITY; -- partially_plated_interconnect_module_edge

```

```

ENTITY partially_routed_physical_network
    SUBTYPE OF (physical_network);
        unrouted_connectivity_requirement_element : SET[1:?] OF unrouted_join_relationship;
END_ENTITY; -- partially_routed_physical_network

```

```

ENTITY participant_date_and_time_assignment
    SUBTYPE OF (managed_design_object);
        action_participant : person_and_organization;
        action_date_and_time : date_and_time;
END_ENTITY; -- participant_date_and_time_assignment

```

```

ENTITY passage_technology
  SUPERTYPE OF (ONEOF (default_via_definition,
default_component_termination_passage_definition,
default_unsupported_passage_definition))
  SUBTYPE OF (managed_design_object);
    associated_definition                : interconnect_module;
    as_finished_inter_stratum_extent     : inter_stratum_extent;
    deposition_material                  : OPTIONAL ee_material;
    plated_passage                       : BOOLEAN;
    maximum_as_finished_passage_extent   : OPTIONAL length_data_element;
    minimum_as_finished_passage_extent    : OPTIONAL length_data_element;
    name                                 : ee_name;
    maximum_as_finished_deposition_thickness : OPTIONAL length_data_element;
    maximum_aspect_ratio                 : OPTIONAL REAL;
    minimum_as_finished_deposition_thickness : OPTIONAL length_data_element;
    specification                        : OPTIONAL process_specification;
  UNIQUE
    UR1 : name,
associated_definition;
END_ENTITY; -- passage_technology

```

```

ENTITY pca
  SUBTYPE OF (assembly_module);
    SELF\physical_unit_design_view.usage_view : pca_usage_view;
END_ENTITY; -- pca

```

```

ENTITY pca_terminal
  SUBTYPE OF (assembly_module_terminal);
    SELF\part_feature.associated_definition : pca_usage_view;
END_ENTITY; -- pca_terminal

```

```

ENTITY pca_usage_view
  SUBTYPE OF (assembly_module_usage_view);
END_ENTITY; -- pca_usage_view

```

```

ENTITY pcb
  SUPERTYPE OF (layout_macro_definition)
  SUBTYPE OF (interconnect_module);
    SELF\physical_unit_design_view.usage_view : OPTIONAL pcb_usage_view;
  WHERE
    WR1 : ('AP210_ARM.LAYOUT_MACRO_DEFINITION' IN TYPEOF(SELF)) XOR (EXISTS(SELF\physical_unit_design_view.usage_view));
END_ENTITY; -- pcb

```

```

ENTITY pcb_usage_view

```

```

    SUBTYPE OF (interconnect_module_usage_view);
END_ENTITY; -- pcb_usage_view

```

```

ENTITY perpendicular_shape_element_relationship
    SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- perpendicular_shape_element_relationship

```

```

ENTITY perpendicularity_tolerance
    SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- perpendicularity_tolerance

```

```

ENTITY person
    SUBTYPE OF (managed_design_object);
    identifier : ee_name;
    address    : OPTIONAL STRING;
END_ENTITY; -- person

```

```

ENTITY person_and_organization
    SUBTYPE OF (managed_design_object);
    assigned_person      : person;
    assigned_organization : organization;
    person_id_in_organization : OPTIONAL ee_name;
    UNIQUE
        UR1 : assigned_person,
assigned_organization;
        UR2 : person_id_in_organization,
assigned_organization;
END_ENTITY; -- person_and_organization

```

```

ENTITY physical_component
    SUPERTYPE OF (ONEOF (cable_component,
packaged_component,
interconnect_module_component,
assembly_module_component,
bare_die_component,
routed_physical_component,
routed_interconnect_component) ANDOR
interface_component)
    SUBTYPE OF (assembly_component);
    SELF\assembly_component.usage_definition : OPTIONAL physical_unit;
END_ENTITY; -- physical_component

```

```

ENTITY physical_component_feature
    ABSTRACT SUPERTYPE OF (component_mounting_feature ANDOR

```

```

interconnect_module_component_surface_feature ANDOR
(ONEOF (physical_component_terminal,
component_interface_terminal)))
    SUBTYPE OF (component_feature,
physical_feature_or_part_template);
END_ENTITY; -- physical_component_feature

```

```

ENTITY physical_component_terminal
    ABSTRACT SUPERTYPE OF (ONEOF (assembly_module_component_terminal,
bare_die_component_terminal,
interconnect_component_join_terminal,
interconnect_module_component_terminal,
minimally_defined_component_terminal,
packaged_component_join_terminal))
    SUBTYPE OF (component_terminal,
physical_component_feature);
    INVERSE
        connected_node : SET[0:1] OF physical_connectivity_definition FOR associated_terminal;
END_ENTITY; -- physical_component_terminal

```

```

ENTITY physical_connectivity_abstraction_map
    SUBTYPE OF (managed_design_object);
    name_space_context_based_definition : physical_connectivity_element;
    geometric_context_based_definition : conductive_interconnect_element;
    UNIQUE
        UR1 : name_space_context_based_definition,
geometric_context_based_definition;
END_ENTITY; -- physical_connectivity_abstraction_map

```

```

ENTITY physical_connectivity_definition
    SUPERTYPE OF (physical_connectivity_structure_definition)
    SUBTYPE OF (managed_design_object);
    associated_definition : physical_unit_network_definition;
    associated_terminals : SET[2:?] OF physical_component_terminal;
    name : ee_name;
    UNIQUE
        UR1 : name,
associated_definition;
END_ENTITY; -- physical_connectivity_definition

```

```

ENTITY physical_connectivity_definition_relationship
    SUBTYPE OF (managed_design_object);
    requirement : ee_requirement_occurrence;
    previous_definition : physical_connectivity_definition;
    current_definition : physical_connectivity_definition;
END_ENTITY; -- physical_connectivity_definition_relationship

```

```

ENTITY physical_connectivity_element
  SUBTYPE OF (managed_design_object);
    start_terminus      : termination_or_junction;
    name                 : ee_name;
    end_terminus         : termination_or_junction;
    connectivity_context : physical_connectivity_structure_definition;
  UNIQUE
    UR1 : name,
connectivity_context;
END_ENTITY; -- physical_connectivity_element

```

```

ENTITY physical_connectivity_interrupting_cutout
  SUBTYPE OF (cutout);
    interrupted_connectivity_element : SET[1:?] OF conductive_interconnect_element;
END_ENTITY; -- physical_connectivity_interrupting_cutout

```

```

ENTITY physical_connectivity_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_definition : physical_connectivity_definition;
    resultant_definition : physical_connectivity_definition;
END_ENTITY; -- physical_connectivity_make_from_relationship

```

```

ENTITY physical_connectivity_structure_definition
  SUBTYPE OF (physical_connectivity_definition);
  INVERSE
    structural_element : SET[1:?] OF physical_connectivity_element FOR connectivity_cont
END_ENTITY; -- physical_connectivity_structure_definition

```

```

ENTITY physical_feature_or_part_template
  ABSTRACT SUPERTYPE OF (ONEOF (part_feature,
physical_component_feature,
physical_laminate_component,
part_template,
stratum_feature) ANDOR
ONEOF (datum_feature,
datum_target))
  SUBTYPE OF (shape_element);
END_ENTITY; -- physical_feature_or_part_template

```

```

ENTITY physical_laminate_component
  SUBTYPE OF (laminate_component,
physical_feature_or_part_template);
  WHERE
    WR1 : SIZEOF(TYPEOF(SELF)) > 6;
END_ENTITY; -- physical_laminate_component

```

```

ENTITY physical_network
  SUPERTYPE OF (partially_routed_physical_network)
  SUBTYPE OF (managed_design_object);
    current_associated_definition          : interconnect_module;
    prior_associated_definition            : OPTIONAL SET[1:?] OF interconnect_mod
    connectivity_requirement_element        : SET[2:?] OF laminate_component_join_t
    interconnect_module_network_topology_element : SET[1:?] OF join_relationship;
    name                                   : ee_name;
    reference_connected_terminals          : OPTIONAL part_connected_terminals_def
  UNIQUE
    UR1 : name,
current_associated_definition;
  WHERE
    WR1 : NOT (EXISTS(prior_associated_definition)) OR NOT (current_associated_definition
    WR2 : SIZEOF(QUERY(cre <* connectivity_requirement_element | NOT (NOT ('AP210_ARM.LA
'AP210_ARM.' + 'COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_LAND',
'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND']) = 1))) = 0;
    WR3 : SIZEOF(QUERY(cre <* connectivity_requirement_element | NOT (NOT ('AP210_ARM.NO
'AP210_ARM.' + 'COMPONENT_TERMINATION_PASSAGE_AND_CONTACT_SIZE_DEPENDENT_NON_FUNCTIONAL_L
'AP210_ARM.VIA_AND_CONTACT_SIZE_DEPENDENT_LAND']) = 1))) = 0;
  END_ENTITY; -- physical_network

```

```

ENTITY physical_network_supporting_inter_stratum_feature
  SUPERTYPE OF (join_two_physical_connectivity_definition_supporting_inter_stratum_featu
  SUBTYPE OF (plated_inter_stratum_feature,
inter_stratum_join_implementation);
  END_ENTITY; -- physical_network_supporting_inter_stratum_feature

```

```

ENTITY physical_network_supporting_stratum_feature_conductive_join
  SUBTYPE OF (stratum_feature_conductive_join,
inter_stratum_join_implementation);
  END_ENTITY; -- physical_network_supporting_stratum_feature_conductive_join

```

```

ENTITY physical_node_requirement_to_implementing_component_allocation
  SUBTYPE OF (managed_design_object);
    physical_node_requirement          : physical_connectivity_definition;
    implementation_component            : assembly_component;
    implementation_interface_joint_set : OPTIONAL SET[1:?] OF interface_mounted_join;
    implementation_joint_set           : OPTIONAL SET[1:?] OF assembly_joint;
  DERIVE
    implementation_connection_definition_1 : SET[0:1] OF part_connected_terminals_defini
implementation_joint_set);
    implementation_connection_definition_2 : SET[0:1] OF packaged_connector_terminal_rel
implementation_interface_joint_set);
  WHERE
    WR1 : EXISTS(implementation_interface_joint_set) XOR EXISTS(implementation_joint_set
    WR2 : (SIZEOF(implementation_connection_definition_1) + SIZEOF(implementation_conne

```

```
implementation_joint_set)))) OR ('AP210_ARM.ROUTED_INTERCONNECT_COMPONENT' IN TYPEOF(impl
    WR3 : SIZEOF(QUERY(pct <* physical_node_requirement.associated_terminals | pct\compon
END_ENTITY; -- physical_node_requirement_to_implementing_component_allocation
```

```
ENTITY physical_unit
  ABSTRACT SUPERTYPE OF (ONEOF (physical_unit_design_view,
physical_unit_usage_view))
  SUBTYPE OF (ee_product_definition);
    analytical_physical_unit_representation : OPTIONAL SET[1:?] OF analytical_represen
    unit_technology : OPTIONAL technology;
    physical_device_characteristic : OPTIONAL SET[1:?] OF characteristic;
    physical_device_coordinated_characteristic : OPTIONAL SET[1:?] OF coordinated_charac
    design_requirement : OPTIONAL SET[1:?] OF ee_requirement_occurrence;
END_ENTITY; -- physical_unit
```

```
ENTITY physical_unit_3d_keepout_shape
  SUBTYPE OF (bound_volume_shape,
non_feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    constrained_design_object_category : keepout_product_design_object_category;
    shape_characterized_physical_unit : physical_unit;
    shape_material_condition : material_condition;
    shape_purpose : keepout_purpose;
    shape_environment : environment;
    centroid_location : OPTIONAL cartesian_point;
END_ENTITY; -- physical_unit_3d_keepout_shape
```

```
ENTITY physical_unit_3d_shape
  SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
    application_technology_constraint : OPTIONAL ee_requirement_occurrence;
    shape_characterized_physical_unit : physical_unit;
    shape_material_condition : material_condition;
    shape_purpose : three_d_purpose;
    shape_environment : environment;
    centroid_location : OPTIONAL cartesian_point;
  WHERE
    WR1 : NOT ('AP210_ARM.PACKAGED_PART' IN TYPEOF(shape_characterized_physical_unit));
END_ENTITY; -- physical_unit_3d_shape
```

```
ENTITY physical_unit_design_view
  SUPERTYPE OF (ONEOF (assembly_module,
interconnect_module))
  SUBTYPE OF (physical_unit);
    usage_view : OPTIONAL physical_unit_usage_view;
  WHERE
    WR1 : ('AP210_ARM.LAYOUT_MACRO_DEFINITION' IN TYPEOF(SELF)) OR EXISTS(usage_view);
```



```
END_ENTITY; -- physical_unit_design_view
```

```
ENTITY physical_unit_network_definition
  SUBTYPE OF (ee_product_definition);
END_ENTITY; -- physical_unit_network_definition
```

```
ENTITY physical_unit_planar_keepout_shape
  SUBTYPE OF (planar_shape,
non_feature_shape_definition);
  application_technology_constraint      : OPTIONAL ee_requirement_occurrence;
  constrained_design_object_category     : keepout_product_design_object_category;
  shape_characterized_physical_unit      : physical_unit;
  shape_distance_from_seating_plane      : OPTIONAL length_data_element;
  shape_location_with_respect_to_seating_plane : shape_location;
  shape_material_condition               : material_condition;
  centroid_location                     : OPTIONAL cartesian_point;
  shape_environment                     : environment;
  shape_purpose                           : keepout_purpose;
  WHERE
    WR1 : NOT ((congruent = shape_location_with_respect_to_seating_plane) AND (EXISTS(shape_location_with_respect_to_seating_plane)));
END_ENTITY; -- physical_unit_planar_keepout_shape
```

```
ENTITY physical_unit_planar_shape
  SUBTYPE OF (planar_projected_shape);
  application_technology_constraint      : OPTIONAL ee_requirement_occurrence;
  shape_characterized_physical_unit      : physical_unit;
  shape_distance_from_seating_plane      : OPTIONAL length_data_element;
  shape_location_with_respect_to_seating_plane : shape_location;
  shape_material_condition               : material_condition;
  centroid_location                     : OPTIONAL cartesian_point;
  shape_environment                     : environment;
  shape_purpose                           : planar_purpose;
  WHERE
    WR1 : NOT ('AP210_ARM.PACKAGED_PART' IN TYPEOF(shape_characterized_physical_unit));
END_ENTITY; -- physical_unit_planar_shape
```

```
ENTITY physical_unit_relationship
  SUBTYPE OF (managed_design_object);
  relating_unit : physical_unit;
  related_unit  : physical_unit;
END_ENTITY; -- physical_unit_relationship
```

```
ENTITY physical_unit_usage_view
  SUPERTYPE OF (ONEOF (package,
bare_die,
```

```

packaged_part,
assembly_module_usage_view,
interconnect_module_usage_view))
    SUBTYPE OF (physical_unit);
END_ENTITY; -- physical_unit_usage_view

```

```

ENTITY pitch_class
    SUBTYPE OF (managed_design_object);
    class_member : range_characteristic;
    class_name   : ee_name;
    WHERE
        WR1 : SIZEOF(QUERY(np <* class_member\coordinated_characteristic.component_characteristic)) > 0;
END_ENTITY; -- pitch_class

```

```

ENTITY placement_group
    SUBTYPE OF (managed_design_object);
    design_specific_usage : ee_name;
    requirement           : SET[1:?] OF ee_requirement_occurrence;
END_ENTITY; -- placement_group

```

```

ENTITY placement_group_area_assignment
    SUBTYPE OF (managed_design_object);
    placed_group : placement_group;
    area         : mounting_restriction_area;
END_ENTITY; -- placement_group_area_assignment

```

```

ENTITY placement_group_volume_assignment
    SUBTYPE OF (managed_design_object);
    placed_group : placement_group;
    volume       : mounting_restriction_volume;
END_ENTITY; -- placement_group_volume_assignment

```

```

ENTITY planar_projected_shape
    SUPERTYPE OF (ONEOF ((assembly_component_2d_shape ANDOR
(ONEOF (part_template_planar_shape,
footprint_definition_shape,
padstack_definition_shape,
physical_unit_planar_shape))),
design_intent_modification_planar_shape,
stratum_feature_planar_shape,
stratum_planar_shape))
    SUBTYPE OF (planar_shape,
feature_shape_definition);
    INVERSE
        curve_basis_1 : SET[0:1] OF volume_shape_plane_projection FOR surface_projection_res

```

```

        curve_basis_2 : SET[0:1] OF volume_shape_intersection FOR surface_intersection_result;
    WHERE
        WR1 : EXISTS(curve_basis_1) XOR EXISTS(curve_basis_2);
END_ENTITY; -- planar_projected_shape

```

```

ENTITY planar_shape
    ABSTRACT SUPERTYPE OF (ONEOF (geometrically_bounded_2d_wireframe,
    shell_based_2d_wireframe,
    edge_based_2d_wireframe,
    csg_2d_shape) ANDOR
    (ONEOF (planar_projected_shape,
    physical_unit_planar_keepout_shape)))
    SUBTYPE OF (shape_definition);
    WHERE
        WR1 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- planar_shape

```

```

ENTITY planned_characteristic
    SUBTYPE OF (managed_design_object);
    reference_requirement : requirement_allocation_select;
    planned_coordinated_characteristic : coordinated_characteristic;
END_ENTITY; -- planned_characteristic

```

```

ENTITY planned_date_effectivity
    SUBTYPE OF (planned_effectivity);
    start_date : date_or_date_and_time;
    end_date : OPTIONAL date_or_date_and_time;
END_ENTITY; -- planned_date_effectivity

```

```

ENTITY planned_effectivity
    SUPERTYPE OF (ONEOF (planned_date_effectivity,
    planned_serial_number_effectivity,
    planned_lot_effectivity))
    SUBTYPE OF (managed_design_object);
    effective_assembly : assembly_composition_relationship;
    effectivity_approval : ee_approval;
    configuration : ee_product_configuration;
    component_quantity : OPTIONAL data_element;
    UNIQUE
    UR1 : effective_assembly,
    configuration;
END_ENTITY; -- planned_effectivity

```

```

ENTITY planned_lot_effectivity
    SUBTYPE OF (planned_effectivity);

```

```

        lot_number : STRING;
        lot_size   : data_element;
END_ENTITY; -- planned_lot_effectivity

```

```

ENTITY planned_serial_number_effectivity
  SUBTYPE OF (planned_effectivity);
  through_effectivity_id : OPTIONAL STRING;
  from_effectivity_id    : STRING;
END_ENTITY; -- planned_serial_number_effectivity

```

```

ENTITY plated_conductive_base_blind_via
  SUBTYPE OF (blind_via);
END_ENTITY; -- plated_conductive_base_blind_via

```

```

ENTITY plated_cutout
  SUBTYPE OF (cutout,
plated_inter_stratum_feature);
END_ENTITY; -- plated_cutout

```

```

ENTITY plated_cutout_edge_segment
  SUBTYPE OF (cutout_edge_segment,
plated_inter_stratum_feature);
  SELF\cutout_edge_segment.composed_cutout : partially_plated_cutout;
END_ENTITY; -- plated_cutout_edge_segment

```

```

ENTITY plated_inter_stratum_feature
  ABSTRACT SUPERTYPE OF (ONEOF (plated_cutout,
plated_interconnect_module_edge,
plated_cutout_edge_segment,
plated_interconnect_module_edge_segment,
plated_passage) ANDOR
physical_network_supporting_inter_stratum_feature)
  SUBTYPE OF (inter_stratum_feature);
END_ENTITY; -- plated_inter_stratum_feature

```

```

ENTITY plated_interconnect_module_edge
  SUBTYPE OF (interconnect_module_edge,
plated_inter_stratum_feature);
END_ENTITY; -- plated_interconnect_module_edge

```

```

ENTITY plated_interconnect_module_edge_segment
  SUBTYPE OF (plated_inter_stratum_feature,

```

```

interconnect_module_edge_segment);
    SELF\interconnect_module_edge_segment.composed_edge : partially_plated_interconnect_
END_ENTITY; -- plated_interconnect_module_edge_segment

```

```

ENTITY plated_passage
    ABSTRACT SUPERTYPE OF (ONEOF (component_termination_passage,
via))
    SUBTYPE OF (plated_inter_stratum_feature);
END_ENTITY; -- plated_passage

```

```

ENTITY plus_minus_tolerance_characteristic
    SUBTYPE OF (tolerance_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET[3:3] OF numeric_param
END_ENTITY; -- plus_minus_tolerance_characteristic

```

```

ENTITY plus_minus_tolerance_range_representation
    SUBTYPE OF (shape_dimension_representation);
    basic_dimension_value : data_element;
    upper_deviation       : data_element;
    lower_deviation       : data_element;
END_ENTITY; -- plus_minus_tolerance_range_representation

```

```

ENTITY polarity_indication_feature
    SUBTYPE OF (part_feature);
    associated_body_vertical_extent : OPTIONAL SET[1:2] OF package_body_surface;
    associated_terminal             : package_terminal;
END_ENTITY; -- polarity_indication_feature

```

```

ENTITY polygonal_area
    SUBTYPE OF (primitive_2d);
    bounds : LIST[3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- polygonal_area

```

```

ENTITY position_tolerance
    ABSTRACT SUPERTYPE OF (ONEOF (position_tolerance_with_referenced_datum_system,
position_tolerance_without_referenced_datum_system) ANDOR
composite_feature_relating_position_tolerance)
    SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- position_tolerance

```

```

ENTITY position_tolerance_with_referenced_datum_system
    SUPERTYPE OF (ONEOF (composite_feature_locating_position_tolerance,

```

```

boundary_based_position_tolerance))
    SUBTYPE OF (position_tolerance,
geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- position_tolerance_with_referenced_datum_system

```

```

ENTITY position_tolerance_without_referenced_datum_system
    SUBTYPE OF (position_tolerance,
geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- position_tolerance_without_referenced_datum_system

```

```

ENTITY positional_boundary
    ABSTRACT SUPERTYPE OF (ONEOF (profile_related_positional_boundary,
dimension_related_positional_boundary))
    SUBTYPE OF (non_feature_shape_element);
    INVERSE
        boundary_definition : positional_boundary_definition FOR defined_positional_boundary
END_ENTITY; -- positional_boundary

```

```

ENTITY positional_boundary_and_profile_boundary_member_relationship
    SUBTYPE OF (managed_design_object);
        offset_positional_boundary : profile_related_positional_boundary;
        base_profile_tolerance_zone_boundary_member : tolerance_zone_boundary_member;
    INVERSE
        offset : positional_boundary_offset FOR offset_boundary_relationship;
END_ENTITY; -- positional_boundary_and_profile_boundary_member_relationship

```

```

ENTITY positional_boundary_definition
    ABSTRACT SUPERTYPE OF (ONEOF (profile_related_positional_boundary_definition,
dimension_related_positional_boundary_definition))
    SUBTYPE OF (managed_design_object);
        defined_positional_boundary : positional_boundary;
END_ENTITY; -- positional_boundary_definition

```

```

ENTITY positional_boundary_member
    SUBTYPE OF (non_feature_shape_element);
    INVERSE
        composed_positional_boundary : dimension_related_positional_boundary FOR boundary_member_definition;
        boundary_member_definition : positional_boundary_member_definition FOR defined_positional_boundary;
END_ENTITY; -- positional_boundary_member

```

```

ENTITY positional_boundary_member_definition
    SUBTYPE OF (managed_design_object);
        defined_positional_boundary_member : positional_boundary_member;
        tolerance_value : length_data_element;

```

```

        associated_dimension          : dimensional_size;
    INVERSE
        boundary_definition : dimension_related_positional_boundary_definition FOR character;
END_ENTITY; -- positional_boundary_member_definition

```

```

ENTITY positional_boundary_offset
    SUBTYPE OF (managed_design_object);
        offset_boundary_relationship : positional_boundary_and_profile_boundary_member_relationship;
        tolerance_value             : length_data_element;
    INVERSE
        boundary_definition : profile_related_positional_boundary_definition FOR boundary_offset;
END_ENTITY; -- positional_boundary_offset

```

```

ENTITY power_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- power_data_element

```

```

ENTITY pre_defined_datum_axis_3d_symbol
    SUBTYPE OF (bound_volume_shape,
non_feature_shape_definition);
        SELF\non_feature_shape_definition.associated_element : datum_axis;
    WHERE
        WR1 : NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
        WR2 : SELF\shape_definition.geometric_context.dimensionality = three_dimensional;
END_ENTITY; -- pre_defined_datum_axis_3d_symbol

```

```

ENTITY pre_defined_datum_plane_3d_symbol
    SUBTYPE OF (bound_volume_shape,
non_feature_shape_definition);
        SELF\non_feature_shape_definition.associated_element : datum_plane;
    WHERE
        WR1 : NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
        WR3 : SELF\shape_definition.geometric_context.dimensionality = three_dimensional;
END_ENTITY; -- pre_defined_datum_plane_3d_symbol

```

```

ENTITY pre_defined_dimension_qualifier
    SUBTYPE OF (dimension_qualifier);
        description : dimension_qualifiers;
END_ENTITY; -- pre_defined_dimension_qualifier

```

```

ENTITY pre_defined_geometric_tolerance_qualifier
    SUBTYPE OF (geometric_tolerance_qualifier);
        description : geometric_tolerance_qualifiers;
END_ENTITY; -- pre_defined_geometric_tolerance_qualifier

```

```

ENTITY pre_defined_parallel_datum_axis_2d_symbol
  SUBTYPE OF (planar_shape,
non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
  WHERE
    WR1 : NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
    WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_parallel_datum_axis_2d_symbol

```

```

ENTITY pre_defined_parallel_datum_axis_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_parallel_da
  SELF\volume_shape_intersection.projected_volume           : pre_defined_datum_axis_
END_ENTITY; -- pre_defined_parallel_datum_axis_symbol_3d_2d_relationship

```

```

ENTITY pre_defined_perpendicular_datum_axis_2d_symbol
  SUBTYPE OF (planar_shape,
non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
  WHERE
    WR1 : NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
    WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_perpendicular_datum_axis_2d_symbol

```

```

ENTITY pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_perpendicul
  SELF\volume_shape_intersection.projected_volume           : pre_defined_datum_axis_
END_ENTITY; -- pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship

```

```

ENTITY pre_defined_perpendicular_datum_plane_2d_symbol
  SUBTYPE OF (planar_shape,
non_feature_shape_definition);
  SELF\non_feature_shape_definition.associated_element : datum_axis;
  WHERE
    WR1 : NOT EXISTS(SELF\non_feature_shape_definition.model_shape);
    WR2 : SELF\shape_definition.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- pre_defined_perpendicular_datum_plane_2d_symbol

```

```

ENTITY pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship
  SUBTYPE OF (volume_shape_intersection);
  SELF\volume_shape_intersection.surface_intersection_result : pre_defined_perpendicul
  SELF\volume_shape_intersection.projected_volume           : pre_defined_datum_plane
END_ENTITY; -- pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship

```



```

ENTITY precision_qualifier
  SUBTYPE OF (managed_design_object);
    precision_value : INTEGER;
END_ENTITY; -- precision_qualifier

```

```

ENTITY pressure_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- pressure_data_element

```

```

ENTITY primary_orientation_feature
  SUBTYPE OF (part_feature,
datum_feature);
    associated_body_vertical_extent : OPTIONAL SET[1:2] OF package_body_surface;
END_ENTITY; -- primary_orientation_feature

```

```

ENTITY primary_reference_terminal
  SUBTYPE OF (package_terminal);
  WHERE
    WR1 : NOT (EXISTS(SELF\part_feature.precedent_feature));
END_ENTITY; -- primary_reference_terminal

```

```

ENTITY primary_stratum_indicator_symbol
  SUBTYPE OF (special_symbol_laminate_component);
END_ENTITY; -- primary_stratum_indicator_symbol

```

```

ENTITY primitive_2d
  ABSTRACT SUPERTYPE OF (ONEOF (circular_area,
elliptic_area,
rectangular_area,
polygonal_area))
  SUBTYPE OF (shape_item);
  WHERE
    WR1 : SELF\shape_item.geometric_context.dimensionality = two_dimensional;
END_ENTITY; -- primitive_2d

```

```

ENTITY printed_component
  SUPERTYPE OF (ONEOF (layout_macro_component,
printed_connector_component,
routed_printed_component))
  SUBTYPE OF (assembly_component);
    cad_layer : OPTIONAL external_definition;
    stratum_feature_implementation : OPTIONAL SET[1:?] OF stratum_feature;
    SELF\assembly_component.usage_definition : printed_part_template_or_printed_part_cro

```

```

        printed_component_stack          : OPTIONAL SET[1:?] OF printed_component_link;
        required_material_stack          : OPTIONAL layer;
    DERIVE
        access_mechanism : SET[1:?] OF printed_component_join_terminal := raw_access_mechanism;
    INVERSE
        raw_access_mechanism      : SET[1:?] OF printed_component_join_terminal FOR associated_precedent_link;
        associated_precedent_link : SET[0:1] OF printed_component_link FOR precedent_printed_component;
        associated_subsequent_link : SET[0:1] OF printed_component_link FOR subsequent_printed_component;
    WHERE
        WR1 : (NOT EXISTS(stratum_feature_implementation) OR same_stratum(stratum_feature_implementation, self));
        WR2 : (NOT ((EXISTS(SELF\assembly_component.usage_definition.of_stratum_technology)) OR EXISTS(SELF\assembly_component.usage_definition.of_stratum_material_stack)));
        WR3 : (NOT ((EXISTS(SELF\assembly_component.usage_definition.material_stack)) XOR EXISTS(SELF\assembly_component.usage_definition.printed_part_template_stack)));
        WR4 : (NOT ((EXISTS(SELF\assembly_component.usage_definition.printed_part_template_stack) OR EXISTS(SELF\assembly_component.usage_definition.printed_part_cross_section_template)) XOR EXISTS(SELF\assembly_component.usage_definition.printed_component_stack)));
        WR5 : (NOT (EXISTS(printed_component_stack)) OR consistent_stack_definition(SELF\assembly_component.usage_definition.printed_component_stack));
        WR6 : (NOT (EXISTS(associated_precedent_link) OR EXISTS(associated_subsequent_link)));
        WR7 : NOT ('AP210_ARM.ROUTED_PRINTED_COMPONENT' IN TYPEOF(SELF)) OR ('AP210_ARM.PRINTED_PART_CROSS_SECTION_TEMPLATE' IN TYPEOF(SELF\assembly_component.usage_definition.printed_part_cross_section_template));
        WR8 : NOT ('AP210_ARM.PRINTED_PART_CROSS_SECTION_TEMPLATE' IN TYPEOF(SELF\assembly_component.usage_definition.printed_part_cross_section_template));
    END_ENTITY; -- printed_component

ENTITY printed_component_join_terminal
    SUBTYPE OF (laminate_component_join_terminal);
    SELF\component_feature.definition      : printed_part_template_join_terminal;
    SELF\component_feature.associated_component : printed_component;
    stratum_concept_implementation        : stratum_feature;
    WHERE
        WR1 : SIZEOF(QUERY(sf <* SELF\component_feature.associated_component\printed_component_join_terminal)) <= 1;
    END_ENTITY; -- printed_component_join_terminal

ENTITY printed_component_link
    SUBTYPE OF (managed_design_object);
    precedent_printed_component : printed_component;
    subsequent_printed_component : printed_component;
    definition                  : printed_part_template_link;
    INVERSE
        associated_printed_component : printed_component FOR printed_component_stack;
    WHERE
        WR1 : precedent_printed_component\assembly_component.usage_definition = definition.printed_component;
        WR2 : subsequent_printed_component\assembly_component.usage_definition = definition.printed_component;
        WR3 : precedent_printed_component <> subsequent_printed_component;
    END_ENTITY; -- printed_component_link

ENTITY printed_connector_component
    SUBTYPE OF (printed_component);
    SELF\assembly_component.usage_definition : printed_connector_template;
    INVERSE
        interface_access_mechanism : SET[1:?] OF printed_connector_component_interface_terminal;
    END_ENTITY; -- printed_connector_component

```

```

ENTITY printed_connector_component_interface_terminal
  SUBTYPE OF (laminate_component_interface_terminal);
  SELF\component_feature.definition          : printed_part_template_interface_terminal;
  SELF\component_feature.associated_component : printed_connector_component;
  stratum_concept_implementation            : stratum_feature;
  WHERE
    WR1 : SIZEOF(QUERY(temp <* SELF\component_feature.associated_component\printed_component)) = 1;
END_ENTITY; -- printed_connector_component_interface_terminal

```

```

ENTITY printed_connector_template
  SUBTYPE OF (printed_part_template);
  INVERSE
    characterized_connector_terminal_relationship : SET[1:?] OF printed_connector_terminal_relationship;
END_ENTITY; -- printed_connector_template

```

```

ENTITY printed_connector_template_terminal_relationship
  SUBTYPE OF (managed_design_object);
  join_terminal          : printed_part_template_join_terminal;
  interface_terminal     : printed_part_template_interface_terminal;
  connector              : printed_connector_template;
  UNIQUE
    UR1 : join_terminal,
connector;
    UR2 : interface_terminal,
connector;
END_ENTITY; -- printed_connector_template_terminal_relationship

```

```

ENTITY printed_part_cross_section_template
  SUBTYPE OF (part_template);
  horizontal_material_link          : OPTIONAL SET[1:?] OF printed_part_template_material_link;
  vertical_material_link           : OPTIONAL SET[1:?] OF printed_part_template_material_link;
  transmission_line_model          : analytical_model;
  maximum_transmission_line_characteristic : OPTIONAL SET[1:?] OF analytical_representation;
  minimum_transmission_line_characteristic : OPTIONAL SET[1:?] OF analytical_representation;
  maximum_width                    : length_data_element;
  minimum_width                    : length_data_element;
  start_vertex                     : edge_segment_vertex;
  end_vertex                       : edge_segment_vertex;
  INVERSE
    using_terminal : SET[1:?] OF printed_part_cross_section_template_terminal FOR cross_section;
  WHERE
    WR1 : SIZEOF(QUERY(hml <* horizontal_material_link | NOT (hml.precedent_material\shape))) = 1;
    WR2 : SIZEOF(QUERY(hml <* horizontal_material_link | NOT (hml.subsequent_material\shape))) = 1;
    WR3 : SIZEOF(QUERY(hml <* vertical_material_link | NOT (hml.precedent_material\shape))) = 1;
    WR4 : SIZEOF(QUERY(hml <* vertical_material_link | NOT (hml.subsequent_material\shape))) = 1;
    WR5 : horizontal_material_link :<>: vertical_material_link;
    WR6 : start_vertex :<>: end_vertex;
END_ENTITY; -- printed_part_cross_section_template

```

```

ENTITY printed_part_cross_section_template_terminal
  SUBTYPE OF (printed_part_template_terminal);
    cross_section_definition      : printed_part_cross_section_template;
    material_to_left_of_terminal  : OPTIONAL printed_part_template_material_link;
    material_to_right_of_terminal : OPTIONAL printed_part_template_material_link;
    material_to_top_of_terminal   : OPTIONAL printed_part_template_material_link;
    material_to_bottom_of_terminal : OPTIONAL printed_part_template_material_link;
  UNIQUE
    UR1 : cross_section_definition,
material_to_left_of_terminal,
material_to_right_of_terminal,
material_to_top_of_terminal,
material_to_bottom_of_terminal;
  WHERE
    WR1 : (SELF\printed_part_template_terminal.connection_zone_category = curve_edge_segmen
    WR2 : NOT (EXISTS(material_to_left_of_terminal) AND EXISTS(material_to_right_of_term
    WR3 : NOT (EXISTS(material_to_left_of_terminal) AND EXISTS(material_to_top_of_termin
    WR4 : NOT (EXISTS(material_to_left_of_terminal) AND EXISTS(material_to_bottom_of_terr
    WR5 : NOT (EXISTS(material_to_right_of_terminal) AND EXISTS(material_to_top_of_termin
    WR6 : NOT (EXISTS(material_to_right_of_terminal) AND EXISTS(material_to_bottom_of_te
    WR7 : NOT (EXISTS(material_to_top_of_terminal) AND EXISTS(material_to_bottom_of_term
END_ENTITY; -- printed_part_cross_section_template_terminal

```

```

ENTITY printed_part_template
  SUPERTYPE OF (ONEOF (layout_macro_floor_plan_template,
printed_connector_template))
  SUBTYPE OF (part_template);
    of_stratum_technology      : OPTIONAL stratum_technology;
    material_stack             : OPTIONAL SET[1:?] OF stratum_technology_link;
    printed_part_template_stack : OPTIONAL SET[1:?] OF printed_part_template_link;
    implemented_function       : functional_unit_usage_view;
  DERIVE
    access_mechanisms : SET[1:?] OF printed_part_template_terminal := raw_access_mechanisms;
  INVERSE
    raw_access_mechanisms : SET[0:?] OF printed_part_template_terminal FOR associated_definition;
  WHERE
    WR1 : ((EXISTS(of_stratum_technology) XOR EXISTS(material_stack)) AND (EXISTS(of_stratum_technology)
    WR2 : (NOT (EXISTS(material_stack)) OR (SIZEOF(QUERY(temp <* material_stack | EXISTS(material_stack)
    WR3 : (SIZEOF(QUERY(prpc <* USEDIN(associated_definition\ee_product_definition.version\AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT')) | (prpc.name = 'printed part'
    WR4 : (EXISTS(printed_part_template_stack) XOR EXISTS(raw_access_mechanisms)) OR ('AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT'))
END_ENTITY; -- printed_part_template

```

```

ENTITY printed_part_template_connected_terminals_definition
  SUBTYPE OF (managed_design_object);
    connected_terminals : SET[2:?] OF printed_part_template_terminal;
    name                 : ee_name;
  UNIQUE
    UR1 : name;

```

```
END_ENTITY; -- printed_part_template_connected_terminals_definition
```

```
ENTITY printed_part_template_interface_terminal
  SUBTYPE OF (printed_part_template_terminal);
  SELF\printed_part_template_terminal.associated_definition : printed_connector_template_terminal_relational;
  INVERSE
    external_connector_function : SET[1:?] OF printed_connector_template_terminal_relational;
END_ENTITY; -- printed_part_template_interface_terminal
```

```
ENTITY printed_part_template_join_terminal
  SUBTYPE OF (printed_part_template_terminal);
  INVERSE
    internal_connector_function : SET[0:1] OF printed_connector_template_terminal_relational;
END_ENTITY; -- printed_part_template_join_terminal
```

```
ENTITY printed_part_template_link
  SUBTYPE OF (stratum_technology_link);
  precedent_printed_part_template : printed_part_template;
  subsequent_printed_part_template : printed_part_template;
  INVERSE
    associated_printed_part_template : printed_part_template FOR printed_part_template_stratum_technology;
  WHERE
    WR1 : EXISTS(precedent_printed_part_template.of_stratum_technology);
    WR2 : EXISTS(subsequent_printed_part_template.of_stratum_technology);
    WR3 : precedent_printed_part_template.of_stratum_technology = SELF\stratum_technology;
    WR4 : subsequent_printed_part_template.of_stratum_technology = SELF\stratum_technology;
END_ENTITY; -- printed_part_template_link
```

```
ENTITY printed_part_template_material
  SUBTYPE OF (managed_design_object);
  maximum_width : OPTIONAL length_data_element;
  minimum_width : OPTIONAL length_data_element;
  required_material : stratum_technology;
  INVERSE
    boundary : SET[0:4] OF template_material_cross_section_boundary FOR associated_definition;
  WHERE
    WR1 : SIZEOF(QUERY(temp <* (USEDIN(SELF,
'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' + 'ASSOCIATED_DEFINITION')) | temp\
    WR2 : SIZEOF(QUERY(temp <* (USEDIN(SELF,
'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' + 'ASSOCIATED_DEFINITION')) | temp\
    WR3 : SIZEOF(QUERY(temp <* (USEDIN(SELF,
'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' + 'ASSOCIATED_DEFINITION')) | temp\
    WR4 : SIZEOF(QUERY(temp <* (USEDIN(SELF,
'AP210_ARM.TEMPLATE_MATERIAL_CROSS_SECTION_BOUNDARY.' + 'ASSOCIATED_DEFINITION')) | temp\
END_ENTITY; -- printed_part_template_material
```

```

ENTITY printed_part_template_material_link
  SUBTYPE OF (managed_design_object);
    precedent_material : template_material_cross_section_boundary;
    subsequent_material : template_material_cross_section_boundary;
  UNIQUE
    UR1 : precedent_material,
subsequent_material;
  WHERE
    WR1 : precedent_material :<>: subsequent_material;
    WR2 : precedent_material\shape_element.name IN ['bottom',
'right'];
    WR3 : subsequent_material\shape_element.name IN ['top',
'left'];
END_ENTITY; -- printed_part_template_material_link

```

```

ENTITY printed_part_template_terminal
  ABSTRACT SUPERTYPE OF (printed_part_cross_section_template_terminal ANDOR
(ONEOF (printed_part_template_join_terminal,
printed_part_template_interface_terminal)))
  SUBTYPE OF (managed_design_object);
    associated_definition : printed_part_template;
    connection_zone_category : printed_part_template_terminal_class;
    terminal_connection_zone : OPTIONAL SET[1:?] OF connection_zone;
    name : ee_name;
  INVERSE
    connection_requirement : SET[0:1] OF printed_part_template_connected_terminal;
    functional_usage_view_terminal : SET[0:1] OF functional_usage_view_to_printed_part_t;
    feature_shape : SET[1:?] OF usage_concept_usage_relationship FOR as
  UNIQUE
    UR1 : name,
associated_definition;
END_ENTITY; -- printed_part_template_terminal

```

```

ENTITY printed_part_template_terminal_relationship
  SUBTYPE OF (managed_design_object);
    related_printed_part_terminal : printed_part_template_terminal;
    relating_printed_part_terminal : printed_part_template_terminal;
END_ENTITY; -- printed_part_template_terminal_relationship

```

```

ENTITY probe_access_area
  SUPERTYPE OF (internal_probe_access_area)
  SUBTYPE OF (laminare_component);
    probed_layout_item : stratum_feature;
    probe_contact_zone : connection_zone;
END_ENTITY; -- probe_access_area

```

```

ENTITY process_specification

```

```

    SUPERTYPE OF (ONEOF (assembly_technology_specification,
fabrication_technology_specification,
lead_form_specification))
    SUBTYPE OF (ee_specification);
END_ENTITY; -- process_specification

```

```

ENTITY product_association
    SUBTYPE OF (managed_design_object);
    relating_product : ee_product_version;
    related_product : SET[1:?] OF ee_product_version;
    constraint_source : ee_specification;
    UNIQUE
        UR1 : constraint_source,
relating_product;
END_ENTITY; -- product_association

```

```

ENTITY profile_boundary_definition
    SUPERTYPE OF (profile_boundary_definition_with_offsets)
    SUBTYPE OF (boundary_zone_definition_with_specified_size);
    basic_profile : true_profile;
    SELF\tolerance_zone_boundary_definition.defined_zone_boundary : tolerance_zone_oppos
END_ENTITY; -- profile_boundary_definition

```

```

ENTITY profile_boundary_definition_with_offsets
    SUBTYPE OF (profile_boundary_definition);
    SELF\tolerance_zone_boundary_definition.defined_zone_boundary : tolerance_zone_expli
    boundary_offsets : SET[2:2] OF tolerance
END_ENTITY; -- profile_boundary_definition_with_offsets

```

```

ENTITY profile_of_any_line_tolerance
    ABSTRACT SUPERTYPE OF (ONEOF (profile_of_any_line_tolerance_with_referenced_datum_syst
profile_of_any_line_tolerance_without_referenced_datum_system) ANDOR
composite_linear_profile_refinement_tolerance)
    SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- profile_of_any_line_tolerance

```

```

ENTITY profile_of_any_line_tolerance_with_referenced_datum_system
    SUPERTYPE OF (composite_linear_profile_locating_tolerance)
    SUBTYPE OF (profile_of_any_line_tolerance,
geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- profile_of_any_line_tolerance_with_referenced_datum_system

```

```

ENTITY profile_of_any_line_tolerance_without_referenced_datum_system
    SUBTYPE OF (profile_of_any_line_tolerance,

```

```

geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- profile_of_any_line_tolerance_without_referenced_datum_system

```

```

ENTITY profile_of_any_surface_tolerance
  ABSTRACT SUPERTYPE OF (ONEOF (profile_of_any_surface_tolerance_with_referenced_datum_s
profile_of_any_surface_tolerance_without_referenced_datum_system) ANDOR
composite_surface_profile_refinement_tolerance)
  SUBTYPE OF (geometric_tolerance);
END_ENTITY; -- profile_of_any_surface_tolerance

```

```

ENTITY profile_of_any_surface_tolerance_with_referenced_datum_system
  SUPERTYPE OF (composite_surface_profile_locating_tolerance)
  SUBTYPE OF (profile_of_any_surface_tolerance,
geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- profile_of_any_surface_tolerance_with_referenced_datum_system

```

```

ENTITY profile_of_any_surface_tolerance_without_referenced_datum_system
  SUBTYPE OF (profile_of_any_surface_tolerance,
geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- profile_of_any_surface_tolerance_without_referenced_datum_system

```

```

ENTITY profile_related_positional_boundary
  SUBTYPE OF (positional_boundary);
  INVERSE
    associating_relationship : positional_boundary_and_profile_boundary_member_relations
END_ENTITY; -- profile_related_positional_boundary

```

```

ENTITY profile_related_positional_boundary_definition
  SUBTYPE OF (positional_boundary_definition);
  SELF\positional_boundary_definition.defined_positional_boundary : profile_related_po
  boundary_offset : positional_boundary
END_ENTITY; -- profile_related_positional_boundary_definition

```

```

ENTITY projected_tolerance_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
  projected_zone_height : projected_zone_height_characteristic;
END_ENTITY; -- projected_tolerance_zone_definition

```

```

ENTITY projected_zone_and_base_relationship
  SUBTYPE OF (managed_design_object);
  base_physical_feature : physical_feature_or_part_template;
  projected_zone : tolerance_zone;

```



```

    INVERSE
        size_characteristic : projected_zone_height_characteristic FOR defined_height_of;
END_ENTITY; -- projected_zone_and_base_relationship

```

```

ENTITY projected_zone_height_characteristic
    SUBTYPE OF (managed_design_object);
        defined_height_of      : projected_zone_and_base_relationship;
        projected_height_value : length_data_element;
    INVERSE
        zone_definition : projected_tolerance_zone_definition FOR projected_zone_height;
END_ENTITY; -- projected_zone_height_characteristic

```

```

ENTITY promissory_usage_in_product_model_relationship
    SUBTYPE OF (managed_design_object);
        component      : ee_product_definition;
        security_code  : OPTIONAL security_classification;
        product_model  : ee_product_model;
END_ENTITY; -- promissory_usage_in_product_model_relationship

```

```

ENTITY promissory_usage_relationship
    SUBTYPE OF (managed_design_object);
        component      : physical_unit;
        security_code  : OPTIONAL security_classification;
        assembly       : physical_unit_design_view;
    UNIQUE
        UR1 : component,
assembly;
END_ENTITY; -- promissory_usage_relationship

```

```

ENTITY purpose_occurrence
    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- purpose_occurrence

```

```

ENTITY qualitative_uncertainty
    SUBTYPE OF (uncertainty_qualifier);
        uncertainty_value : text;
END_ENTITY; -- qualitative_uncertainty

```

```

ENTITY radius_edge_feature_shape
    SUBTYPE OF (edge_shape);
        minimum_radius_length : length_data_element;
        maximum_radius_length : length_data_element;
END_ENTITY; -- radius_edge_feature_shape

```

```

ENTITY range_characteristic
  SUBTYPE OF (coordinated_characteristic);
    SELF\coordinated_characteristic.component_characteristic : SET[2:2] OF numeric_param
    range_type                                              : range_class;
END_ENTITY; -- range_characteristic

```

```

ENTITY ratio_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- ratio_data_element

```

```

ENTITY rectangular_area
  SUBTYPE OF (primitive_2d);
    position : axis_placement_2d;
    x        : length_data_element;
    y        : length_data_element;
END_ENTITY; -- rectangular_area

```

```

ENTITY rectangular_array_placement_group_component
  SUBTYPE OF (array_placement_group);
    element                : SET[1:?] OF linear_array_placement_group_comp
    initial_row_positive_y_displacement : length_data_element;
    terminus_row_positive_y_displacement : length_data_element;
    location_uncertainty      : length_data_element;
END_ENTITY; -- rectangular_array_placement_group_component

```

```

ENTITY reference_composition_path
  SUBTYPE OF (managed_design_object);
    path_element      : SET[1:?] OF functional_unit;
    reference_functional_unit : functional_unit;
END_ENTITY; -- reference_composition_path

```

```

ENTITY reference_functional_unit_assignment_to_part
  SUBTYPE OF (managed_design_object);
    functional_design_definition_path : reference_composition_path;
    implementation                    : physical_unit_or_part_template;
    of_property                       : OPTIONAL SET[1:?] OF coordinated_characteristic;
    path_alias                        : ee_name;
    swap_code                         : OPTIONAL STRING;
  UNIQUE
    UR1 : implementation,
path_alias;
END_ENTITY; -- reference_functional_unit_assignment_to_part

```

```

ENTITY reference_packaged_part_assembly_implementation
  SUBTYPE OF (assembly_module);
  WHERE
    WR1 : SIZEOF(QUERY(prpc <* USEDIN(SELF\ee_product_definition.version.versioned_item,
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | NOT (SIZEOF(QUERY(epcr <* U
'AP210_ARM.EE_PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY') | (epcr.sub_category\ee_product_ca
END_ENTITY; -- reference_packaged_part_assembly_implementation

```

```

ENTITY reference_packaged_part_interconnect_implementation
  SUBTYPE OF (interconnect_module);
  WHERE
    WR1 : SIZEOF(QUERY(prpc <* USEDIN(SELF\ee_product_definition.version.versioned_item,
'AP210_ARM.EE_PRODUCT_RELATED_PRODUCT_CATEGORY.OF_PRODUCT') | NOT (SIZEOF(QUERY(epcr <* U
'AP210_ARM.EE_PRODUCT_CATEGORY_RELATIONSHIP.CATEGORY') | (epcr.sub_category\ee_product_ca
END_ENTITY; -- reference_packaged_part_interconnect_implementation

```

```

ENTITY registered_font
  SUBTYPE OF (managed_design_object);
  registration_document : ee_document;
  font_code             : STRING;
  UNIQUE
  UR1 : font_code,
registration_document;
END_ENTITY; -- registered_font

```

```

ENTITY requirement_allocation
  SUBTYPE OF (managed_design_object);
  target_product_object : SET[1:?] OF product_object_select;
  allocated_requirement : ee_requirement_occurrence;
END_ENTITY; -- requirement_allocation

```

```

ENTITY requirement_composition
  SUBTYPE OF (managed_design_object);
  element      : SET[1:?] OF ee_requirement_occurrence;
  composition : requirement_group_occurrence;
END_ENTITY; -- requirement_composition

```

```

ENTITY requirement_definition
  SUBTYPE OF (ee_product_definition);
  INVERSE
  associated_requirement_occurrence : ee_requirement_occurrence FOR associated_definiti
END_ENTITY; -- requirement_definition

```

```

ENTITY requirement_group_occurrence

```

```

    SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- requirement_group_occurrence

```

```

ENTITY resistance_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- resistance_data_element

```

```

ENTITY restraint_condition
    SUPERTYPE OF (tolerance_specific_restraint_condition)
    SUBTYPE OF (managed_design_object);
    restraint_description : ee_requirement_occurrence;
    applied_to            : restrained_item_select;
END_ENTITY; -- restraint_condition

```

```

ENTITY right_angular_wedge
    SUBTYPE OF (csg_primitive);
END_ENTITY; -- right_angular_wedge

```

```

ENTITY right_circular_cylinder
    SUBTYPE OF (csg_primitive);
END_ENTITY; -- right_circular_cylinder

```

```

ENTITY routed_interconnect_component
    SUBTYPE OF (physical_component);
    routed_centreline_shape : curve;
END_ENTITY; -- routed_interconnect_component

```

```

ENTITY routed_physical_component
    SUBTYPE OF (physical_component);
    routed_centreline_shape : curve;
END_ENTITY; -- routed_physical_component

```

```

ENTITY routed_printed_component
    SUBTYPE OF (printed_component);
    routed_centreline_shape           : curve;
    SELF\assembly_component.usage_definition : printed_part_cross_section_template;
    component_extent                  : inter_stratum_extent;
END_ENTITY; -- routed_printed_component

```

```

ENTITY rule_action
    ABSTRACT SUPERTYPE OF (ONEOF (rule_submission,

```

```

rule_adoption,
rule_rejection,
rule_supersedence,
rule_creation,
rule_expiration,
rule_change_request,
rule_request,
rule_modification))
    SUBTYPE OF (managed_design_object);
        subject_action_assignment : participant_date_and_time_assignment;
        subject_rule                : ee_rule_version;
    UNIQUE
        UR1 : subject_rule,
subject_action_assignment;
END_ENTITY; -- rule_action

```

```

ENTITY rule_adoption
    SUBTYPE OF (rule_action);
END_ENTITY; -- rule_adoption

```

```

ENTITY rule_based_constraint_occurrence
    SUBTYPE OF (constraint_occurrence);
        design_constraint : ee_rule_version;
END_ENTITY; -- rule_based_constraint_occurrence

```

```

ENTITY rule_based_evaluated_characteristic
    SUBTYPE OF (evaluated_characteristic);
        evaluating_rule : ee_rule_version;
END_ENTITY; -- rule_based_evaluated_characteristic

```

```

ENTITY rule_change_request
    SUBTYPE OF (rule_action);
        change_reason : STRING;
END_ENTITY; -- rule_change_request

```

```

ENTITY rule_creation
    SUBTYPE OF (rule_action);
END_ENTITY; -- rule_creation

```

```

ENTITY rule_expiration
    SUBTYPE OF (rule_action);
        expiration_rationale : STRING;
END_ENTITY; -- rule_expiration

```

```

ENTITY rule_justification
  SUBTYPE OF (managed_design_object);
    justified_action      : rule_action;
    justification_rationale : STRING;
END_ENTITY; -- rule_justification

```

```

ENTITY rule_modification
  SUBTYPE OF (rule_action);
    modification_rationale : rule_change_request;
END_ENTITY; -- rule_modification

```

```

ENTITY rule_priority
  SUBTYPE OF (managed_design_object);
    priority      : INTEGER;
    prioritized_rule : ee_rule_definition;
  WHERE
    WR1 : priority >= 0;
END_ENTITY; -- rule_priority

```

```

ENTITY rule_rejection
  SUBTYPE OF (rule_action);
    rejection_reason : STRING;
END_ENTITY; -- rule_rejection

```

```

ENTITY rule_request
  SUBTYPE OF (rule_action);
END_ENTITY; -- rule_request

```

```

ENTITY rule_set
  SUPERTYPE OF (rule_set_group)
  SUBTYPE OF (managed_design_object);
    rule_priority_member : SET[1:?] OF rule_priority;
    identifier           : STRING;
    purpose              : STRING;
  UNIQUE
    UR1 : identifier;
END_ENTITY; -- rule_set

```

```

ENTITY rule_set_group
  SUBTYPE OF (rule_set);
    elements : SET[2:?] OF rule_set;
END_ENTITY; -- rule_set_group

```

```

ENTITY rule_submission
  SUBTYPE OF (rule_action);
  submission_rationale : STRING;
END_ENTITY; -- rule_submission

```

```

ENTITY rule_supersedence
  SUBTYPE OF (rule_action);
  superseded_rule : ee_rule_version;
END_ENTITY; -- rule_supersedence

```

```

ENTITY runout_tolerance_zone_definition_with_specified_angle
  SUBTYPE OF (tolerance_zone_definition);
  runout_orientation : runout_zone_orientation;
END_ENTITY; -- runout_tolerance_zone_definition_with_specified_angle

```

```

ENTITY runout_tolerance_zone_orienting_relationship
  SUBTYPE OF (managed_design_object);
  oriented_zone : tolerance_zone;
  orienting_axis : directed_axis;
  INVERSE
    angular_size_characteristic : runout_zone_orientation FOR oriented_relationship;
END_ENTITY; -- runout_tolerance_zone_orienting_relationship

```

```

ENTITY runout_zone_orientation
  SUBTYPE OF (managed_design_object);
  oriented_relationship : runout_tolerance_zone_orienting_relationship;
  semi_angle_value : angle_data_element;
  INVERSE
    zone_definition : runout_tolerance_zone_definition_with_specified_angle FOR runout_or
END_ENTITY; -- runout_zone_orientation

```

```

ENTITY scalar_terminal_definition
  SUBTYPE OF (functional_unit_usage_view_terminal_definition);
END_ENTITY; -- scalar_terminal_definition

```

```

ENTITY scalar_terminal_definition_link
  SUBTYPE OF (managed_design_object);
  precedent_terminal_definition : scalar_terminal_definition;
  subsequent_terminal_definition : scalar_terminal_definition;
  UNIQUE
    UR1 : precedent_terminal_definition,
subsequent_terminal_definition;
  WHERE

```

```

        WR1 : precedent_terminal_definition :<>: subsequent_terminal_definition;
END_ENTITY; -- scalar_terminal_definition_link

```

```

ENTITY scattering_parameter_data_element
    SUBTYPE OF (data_element);
END_ENTITY; -- scattering_parameter_data_element

```

```

ENTITY seating_plane
    SUBTYPE OF (non_feature_shape_element);
    WHERE
        WR1 : SIZEOF(['AP210_ARM.BARE_DIE',
'AP210_ARM.PACKAGE'] * TYPEOF(SELF\non_feature_shape_element.scope)) = 1;
END_ENTITY; -- seating_plane

```

```

ENTITY secondary_orientation_feature
    SUBTYPE OF (part_feature,
datum_feature);
    associated_body_vertical_extent : OPTIONAL SET[1:2] OF package_body_surface;
END_ENTITY; -- secondary_orientation_feature

```

```

ENTITY security_classification
    SUBTYPE OF (managed_design_object);
    classification_officer          : person_and_organization;
    security_classification_approval : ee_approval;
    classification_date             : date_or_date_and_time;
    classification_level            : ee_name;
    declassification_date           : OPTIONAL date_or_date_and_time;
END_ENTITY; -- security_classification

```

```

ENTITY separate_geometric_tolerance_requirement
    SUBTYPE OF (managed_design_object);
    separated_geometric_tolerances : SET[1:?] OF geometric_tolerance_with_referenced_data;
END_ENTITY; -- separate_geometric_tolerance_requirement

```

```

ENTITY shape_and_ee_product_definition_based_constraint_occurrence
    SUBTYPE OF (ee_product_definition_based_constraint_occurrence);
    constraining_shape : SET[1:?] OF shape_definition;
    WHERE
        WR1 : valid_constraining_shape(constraining_shape,
SELF\ee_product_definition_based_constraint_occurrence.constraining_part);
END_ENTITY; -- shape_and_ee_product_definition_based_constraint_occurrence

```



```

ENTITY shape_composition_relationship
  SUBTYPE OF (geometric_representation_relationship);
    related_feature_shape_placement : axis_placement_or_cto2d;
  DERIVE
    related_feature_shape : non_feature_shape_definition := SELF\geometric_representation_relationship.related_feature_shape_placement;
    resultant_feature_shape : non_feature_shape_definition := SELF\geometric_representation_relationship.related_feature_shape_placement;
END_ENTITY; -- shape_composition_relationship

```

```

ENTITY shape_definition
  ABSTRACT SUPERTYPE OF ((ONEOF (bound_volume_shape,
open_shell_based_surface,
manifold_subsurface_shape,
planar_shape)) AND (ONEOF (feature_shape_definition,
non_feature_shape_definition)))
  SUBTYPE OF (managed_design_object);
    geometric_context : cartesian_coordinate_system;
    element : OPTIONAL SET[1:?] OF shape_item;
END_ENTITY; -- shape_definition

```

```

ENTITY shape_dimension_representation
  ABSTRACT SUPERTYPE OF (ONEOF (limit_dimension_representation,
singular_dimension_representation,
externally_defined_dimension_representation,
plus_minus_tolerance_range_representation))
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- shape_dimension_representation

```

```

ENTITY shape_element
  ABSTRACT SUPERTYPE OF (ONEOF (physical_feature_or_part_template,
component_termination_passage_template_terminal,
land_template_terminal,
via_template_terminal,
non_feature_shape_element) ANDOR
composite_shape_element)
  SUBTYPE OF (managed_design_object);
    name : label;
    description : OPTIONAL label;
  WHERE
    WR1 : NOT ('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) OR (SIZEOF(TYPEOF(SELF)) = 1);
    WR2 : NOT (('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND ('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)));
    WR3 : NOT (('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND ('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)));
    WR4 : NOT (('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)) AND ('AP210_ARM.COMPOSITE_SHAPE_ELEMENT' IN TYPEOF(SELF)));
END_ENTITY; -- shape_element

```

```

ENTITY shape_element_boundary_relationship
  SUBTYPE OF (shape_element_relationship);
END_ENTITY; -- shape_element_boundary_relationship

```

```

ENTITY shape_element_composing_relationship
  SUBTYPE OF (shape_element_constituent_relationship);
  SELF\shape_element_relationship.relate_shape_element : composite_shape_element;
END_ENTITY; -- shape_element_composing_relationship

```

```

ENTITY shape_element_constituent_relationship
  SUPERTYPE OF (shape_element_composing_relationship)
  SUBTYPE OF (shape_element_relationship);
  UNIQUE
    UR1 : related_shape_element,
relate_shape_element;
END_ENTITY; -- shape_element_constituent_relationship

```

```

ENTITY shape_element_deriving_relationship
  SUBTYPE OF (shape_element_relationship);
  SELF\shape_element_relationship.relate_shape_element : derived_shape_element;
  UNIQUE
    UR1 : related_shape_element,
relate_shape_element;
END_ENTITY; -- shape_element_deriving_relationship

```

```

ENTITY shape_element_locating_relationship
  SUBTYPE OF (managed_design_object);
  locate_shape_element : shape_element;
  located_shape_element : shape_element;
  UNIQUE
    UR1 : located_shape_element,
locate_shape_element;
END_ENTITY; -- shape_element_locating_relationship

```

```

ENTITY shape_element_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (parallel_shape_element_relationship,
perpendicular_shape_element_relationship,
datum_difference,
non_feature_shape_element_relationship,
shape_element_boundary_relationship,
shape_element_constituent_relationship,
shape_element_deriving_relationship) ANDOR
datum_feature_usage_in_datum_relationship)
  SUBTYPE OF (managed_design_object);
  relate_shape_element : shape_element;
  description : OPTIONAL label;
  related_shape_element : shape_element;
  name : OPTIONAL label;
END_ENTITY; -- shape_element_relationship

```

```

ENTITY shape_formed_terminal
  SUBTYPE OF (altered_package_terminal);
    terminal_form_specification : lead_form_specification;
END_ENTITY; -- shape_formed_terminal

```

```

ENTITY shape_item
  SUPERTYPE OF ((ONEOF (boolean_csg_result,
    cartesian_point,
    curve,
    axis_placement,
    primitive_2d)) ANDOR
    styled_text)
  SUBTYPE OF (managed_design_object);
  DERIVE
    geometric_context : cartesian_coordinate_system := associated_shape_definition[1].geometric_context;
  INVERSE
    associated_shape_definition : SET[1:?] OF shape_definition FOR element;
  WHERE
    WR1 : NOT (('AP210_ARM.CARTESIAN_POINT' IN TYPEOF(SELF)) AND ('AP210_ARM.STYLED_TEXT' IN TYPEOF(SELF)));
    WR2 : NOT (('AP210_ARM.AXIS_PLACEMENT' IN TYPEOF(SELF)) AND ('AP210_ARM.STYLED_TEXT' IN TYPEOF(SELF)));
END_ENTITY; -- shape_item

```

```

ENTITY shell_based_2d_wireframe
  SUBTYPE OF (planar_shape);
END_ENTITY; -- shell_based_2d_wireframe

```

```

ENTITY shell_based_wireframe
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- shell_based_wireframe

```

```

ENTITY shield
  SUBTYPE OF (assembly_component);
    shielded_item : SET[1:?] OF shieldable_item;
END_ENTITY; -- shield

```

```

ENTITY shield_allocation
  SUBTYPE OF (electromagnetic_compatibility_requirement_allocation);
    SELF\assembly_requirement_allocation.assembly_assignment : shield;
    SELF\assembly_requirement_allocation.requirement : electromagnetic_requirement;
END_ENTITY; -- shield_allocation

```

```

ENTITY signal
  SUBTYPE OF (managed_design_object);

```

```

    signal_category : external_definition;
END_ENTITY; -- signal

```

```

ENTITY signal_property
  SUPERTYPE OF (composite_signal_property)
  SUBTYPE OF (managed_design_object);
  signal_property_representation : characteristic_data_table;
  characterized_signal          : signal;
  signal_property_category      : external_definition;
END_ENTITY; -- signal_property

```

```

ENTITY signal_specification
  SUBTYPE OF (managed_design_object);
  measurement_condition          : test_specification;
  measurement_access_mechanism   : functional_unit_terminal;
  reference_access_mechanism     : OPTIONAL functional_unit_terminal;
  minimum_characteristic         : signal_property;
  maximum_characteristic         : signal_property;
  characterization_network       : functional_unit_network_definition;
  characterizing_functional_unit : functional_unit;
END_ENTITY; -- signal_specification

```

```

ENTITY simultaneous_geometric_tolerance_requirement
  SUBTYPE OF (managed_design_object);
  grouped_geometric_tolerances : SET[2:?] OF geometric_tolerance;
END_ENTITY; -- simultaneous_geometric_tolerance_requirement

```

```

ENTITY single_boundary_dimension
  SUPERTYPE OF (curve_dimension)
  SUBTYPE OF (dimensional_size);
END_ENTITY; -- single_boundary_dimension

```

```

ENTITY single_datum
  SUPERTYPE OF (edge_segment_vertex)
  SUBTYPE OF (datum);
END_ENTITY; -- single_datum

```

```

ENTITY singular_dimension_representation
  ABSTRACT SUPERTYPE OF (ONEOF (theoretically_exact_dimension_representation,
maximum_dimension_representation,
minimum_dimension_representation))
  SUBTYPE OF (shape_dimension_representation);
  dimension_value : data_element;
END_ENTITY; -- singular_dimension_representation

```

```

ENTITY solid_angle_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- solid_angle_data_element

```

```

ENTITY solid_curve
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- solid_curve

```

```

ENTITY solid_fill_style
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- solid_fill_style

```

```

ENTITY solid_of_linear_extrusion
  SUBTYPE OF (bound_volume_shape);
END_ENTITY; -- solid_of_linear_extrusion

```

```

ENTITY special_symbol_laminate_component
  SUPERTYPE OF (primary_stratum_indicator_symbol)
  SUBTYPE OF (laminate_component);
  SELF\assembly_component.usage_definition : special_symbol_part_template;
  stratum_feature_implementation           : stratum_feature;
END_ENTITY; -- special_symbol_laminate_component

```

```

ENTITY special_symbol_part_template
  SUBTYPE OF (part_template);
  of_stratum_technology : stratum_technology;
END_ENTITY; -- special_symbol_part_template

```

```

ENTITY specification_allocation
  SUBTYPE OF (managed_design_object);
  allocated_specification : ee_specification;
  target_product_object  : SET[1:?] OF specification_product_object_select;
END_ENTITY; -- specification_allocation

```

```

ENTITY standard_uncertainty
  SUPERTYPE OF (expanded_uncertainty)
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY; -- standard_uncertainty

```

```

ENTITY start_order
  SUBTYPE OF (work_order);
END_ENTITY; -- start_order

```

```

ENTITY start_request
  SUBTYPE OF (work_request);
END_ENTITY; -- start_request

```

```

ENTITY statistical_dimensional_tolerance
  SUBTYPE OF (dimensional_characteristic);
  alternate_dimension : OPTIONAL dimensional_characteristic;
  statistical_control_requirement : ee_requirement_occurrence;
END_ENTITY; -- statistical_dimensional_tolerance

```

```

ENTITY statistical_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
  alternate_tolerance : OPTIONAL geometric_tolerance;
  statistical_control_requirement : ee_requirement_occurrence;
END_ENTITY; -- statistical_geometric_tolerance

```

```

ENTITY statistical_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
END_ENTITY; -- statistical_tolerance_characteristic

```

```

ENTITY step_edge_feature_shape
  SUBTYPE OF (edge_shape);
  minimum_step_width : length_data_element;
  maximum_step_width : length_data_element;
  maximum_step_depth : length_data_element;
  minimum_step_depth : length_data_element;
END_ENTITY; -- step_edge_feature_shape

```

```

ENTITY straightness_tolerance
  SUBTYPE OF (geometric_tolerance_without_referenced_datum_system);
END_ENTITY; -- straightness_tolerance

```

```

ENTITY stratum
  SUPERTYPE OF (ONEOF (documentation_layer_stratum,
design_layer_stratum))
  SUBTYPE OF (ee_product_definition);
  stratum_usage : OPTIONAL external_definition;
  of_technology : stratum_technology;

```

```

DERIVE
  name : ee_name := SELF\ee_product_definition.product_definition_id;
INVERSE
  of_average_surface : SET[0:1] OF stratum_average_surface FOR of_stratum;
  of_surface          : SET[1:2] OF stratum_surface FOR of_stratum;
  planar_shape        : SET[0:1] OF stratum_planar_shape FOR shape_characterized_stratum;
  assembly            : SET[1:?] OF interconnect_module_stratum_assembly_relationship FOR of_stratum;
  bound_volume_shape  : SET[0:1] OF stratum_3d_shape FOR shape_characterized_stratum;
END_ENTITY; -- stratum

```

```

ENTITY stratum_3d_shape
  SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
  shape_characterized_stratum : stratum;
END_ENTITY; -- stratum_3d_shape

```

```

ENTITY stratum_average_surface
  SUBTYPE OF (managed_design_object);
  of_stratum : stratum;
INVERSE
  of_shape : SET[0:1] OF stratum_average_surface_shape FOR of_surface;
END_ENTITY; -- stratum_average_surface

```

```

ENTITY stratum_average_surface_shape
  SUBTYPE OF (open_shell_based_surface);
  of_surface : stratum_average_surface;
END_ENTITY; -- stratum_average_surface_shape

```

```

ENTITY stratum_concept_relationship
  ABSTRACT SUPERTYPE OF (ONEOF (stratum_feature_conductive_join,
dielectric_crossover_area))
  SUBTYPE OF (managed_design_object);
  resulting_shape : OPTIONAL feature_shape_definition;
  technology_requirement : SET[1:?] OF ee_requirement_occurrence;
END_ENTITY; -- stratum_concept_relationship

```

```

ENTITY stratum_concept_to_physical_usage_view_assignment
  SUBTYPE OF (managed_design_object);
  assigned_design_object : stratum_concept;
  defined_part_feature   : part_feature;
WHERE
  WR1 : NOT ('AP210_ARM.INTERCONNECT_MODULE_TERMINAL' IN TYPEOF(defined_part_feature))
END_ENTITY; -- stratum_concept_to_physical_usage_view_assignment

```

```

ENTITY stratum_feature
  SUPERTYPE OF (ONEOF (conductor,
conductive_filled_area) ANDOR
fiducial_stratum_feature)
  SUBTYPE OF (physical_feature_or_part_template);
    resident_stratum      : stratum;
    SELF\shape_element.name : ee_name;
    feature_of_size       : BOOLEAN;
  UNIQUE
    UR1 : name,
resident_stratum;
END_ENTITY; -- stratum_feature

```

```

ENTITY stratum_feature_3d_shape
  SUBTYPE OF (bound_volume_shape,
feature_shape_definition);
    represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_3d_shape

```

```

ENTITY stratum_feature_conductive_join
  SUPERTYPE OF (physical_network_supporting_stratum_feature_conductive_join)
  SUBTYPE OF (stratum_concept_relationship);
    related_concept : stratum_feature_template_component;
    relating_concept : stratum_feature_template_component;
  INVERSE
    using_via : SET[0:1] OF bonded_conductive_base_blind_via FOR features_join;
  WHERE
    WR1 : related_concept.stratum_feature_implementation.resident_stratum :<>: relating_
END_ENTITY; -- stratum_feature_conductive_join

```

```

ENTITY stratum_feature_in_stratum_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_3d_shape;
    transformation : axis_placement_3d;
    feature_shape : stratum_feature_3d_shape_select;
END_ENTITY; -- stratum_feature_in_stratum_3d_position

```

```

ENTITY stratum_feature_in_stratum_planar_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_planar_shape;
    transformation : axis_placement_2d;
    feature_shape : stratum_feature_planar_representation_select;
END_ENTITY; -- stratum_feature_in_stratum_planar_position

```

```

ENTITY stratum_feature_non_planar_2d_shape

```



```

    SUBTYPE OF (open_shell_based_surface);
    represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_non_planar_2d_shape

```

```

ENTITY stratum_feature_planar_shape
    SUBTYPE OF (planar_projected_shape);
    represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_planar_shape

```

```

ENTITY stratum_feature_shape_stratum_average_surface_shape_relationship
    SUBTYPE OF (managed_design_object);
    of_stratum_feature_shape : stratum_feature_non_planar_2d_shape;
    of_stratum_average_surface_shape : stratum_average_surface_shape;
END_ENTITY; -- stratum_feature_shape_stratum_average_surface_shape_relationship

```

```

ENTITY stratum_feature_shape_stratum_surface_shape_relationship
    SUBTYPE OF (managed_design_object);
    of_stratum_feature_shape : stratum_feature_non_planar_2d_shape;
    of_stratum_surface_shape : stratum_surface_shape;
END_ENTITY; -- stratum_feature_shape_stratum_surface_shape_relationship

```

```

ENTITY stratum_feature_styled_area_representation
    SUBTYPE OF (managed_design_object);
    derived_planar_shape : OPTIONAL SET[1:?] OF stratum_feature_planar_shape;
    shape_styled_area_representation : styled_area_representation;
    represented_stratum_feature : stratum_feature;
END_ENTITY; -- stratum_feature_styled_area_representation

```

```

ENTITY stratum_feature_template
    SUPERTYPE OF (land_physical_template)
    SUBTYPE OF (part_template);
    of_stratum_technology : stratum_technology;
END_ENTITY; -- stratum_feature_template

```

```

ENTITY stratum_feature_template_component
    SUPERTYPE OF (ONEOF (connected_area_component,
non_functional_land,
land))
    SUBTYPE OF (laminate_component);
    SELF\assembly_component.usage_definition : OPTIONAL stratum_feature_template;
    stratum_feature_implementation : stratum_feature;
END_ENTITY; -- stratum_feature_template_component

```

```

ENTITY stratum_interconnect_module_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_component      : interconnect_module_stratum_assembly_relationship;
    stratum_shape          : stratum_3d_shape;
    interconnect_module_shape : physical_unit_3d_shape;
    transformation         : axis_placement_3d;
END_ENTITY; -- stratum_interconnect_module_3d_position

```

```

ENTITY stratum_interconnect_module_planar_position
  SUBTYPE OF (managed_design_object);
    stratum_component      : interconnect_module_stratum_assembly_relationship;
    stratum_shape          : stratum_planar_shape;
    interconnect_module_shape : physical_unit_planar_shape;
    transformation         : axis_placement_2d;
END_ENTITY; -- stratum_interconnect_module_planar_position

```

```

ENTITY stratum_make_from_relationship
  SUBTYPE OF (managed_design_object);
    reusable_stratum : stratum;
    resultant_stratum : stratum;
END_ENTITY; -- stratum_make_from_relationship

```

```

ENTITY stratum_occurrence_specific_padstack_definition
  SUBTYPE OF (padstack_definition);
    reference_stratum : stratum;
END_ENTITY; -- stratum_occurrence_specific_padstack_definition

```

```

ENTITY stratum_planar_shape
  SUBTYPE OF (planar_projected_shape);
    shape_characterized_stratum : stratum;
END_ENTITY; -- stratum_planar_shape

```

```

ENTITY stratum_surface
  SUBTYPE OF (managed_design_object);
    of_stratum : stratum;
    name       : stratum_surface_designation;
  INVERSE
    of_shape : SET[0:1] OF stratum_surface_shape FOR of_surface;
  UNIQUE
    UR1 : name,
of_stratum;
END_ENTITY; -- stratum_surface

```

```

ENTITY stratum_surface_in_stratum_3d_position
  SUBTYPE OF (managed_design_object);
    stratum_shape : stratum_3d_shape;
    transformation : axis_placement_3d;
    surface_shape : stratum_surface_shape_select;
END_ENTITY; -- stratum_surface_in_stratum_3d_position

```

```

ENTITY stratum_surface_shape
  SUBTYPE OF (open_shell_based_surface);
    of_surface : stratum_surface;
END_ENTITY; -- stratum_surface_shape

```

```

ENTITY stratum_technology
  SUPERTYPE OF (ONEOF (design_layer_technology,
documentation_layer_technology))
  SUBTYPE OF (managed_design_object);
    layer_position : layer_position_type;
    stratum_material : ee_material;
    maximum_stratum_thickness : length_data_element;
    minimum_stratum_thickness : length_data_element;
    name : ee_name;
    minimum_finished_feature_size : OPTIONAL length_data_element;
    laminate_stiffness_class : OPTIONAL stiffness_class;
    minimum_finished_feature_spacing : OPTIONAL length_data_element;
    maximum_feature_size_requirement : OPTIONAL length_data_element;
    specification : OPTIONAL process_specification;
  UNIQUE
    UR1 : name;
  WHERE
    WR1 : NOT (layer_position = layer_position_type.primary) OR ('AP210_ARM.DESIGN_LAYER,
    WR2 : NOT (layer_position = layer_position_type.secondary) OR ('AP210_ARM.DESIGN_LAY
END_ENTITY; -- stratum_technology

```

```

ENTITY stratum_technology_link
  SUPERTYPE OF (printed_part_template_link)
  SUBTYPE OF (managed_design_object);
    precedent_stratum_technology : stratum_technology;
    subsequent_stratum_technology : stratum_technology;
    precedent_stratum_technology_link : OPTIONAL stratum_technology_link;
  WHERE
    WR1 : precedent_stratum_technology :<>: subsequent_stratum_technology;
    WR2 : NOT (EXISTS(precedent_stratum_technology_link)) OR (precedent_stratum_technology
END_ENTITY; -- stratum_technology_link

```

```

ENTITY stratum_type_independent_padstack_definition
  SUBTYPE OF (padstack_definition);
END_ENTITY; -- stratum_type_independent_padstack_definition

```

```

ENTITY structured_text
  SUBTYPE OF (managed_design_object);
    content_markup          : external_definition;
    plain_text_content      : OPTIONAL ee_text;
    precedent_structured_text : OPTIONAL structured_text;
    structured_text_content  : OPTIONAL SET[1:?] OF structured_text;
END_ENTITY; -- structured_text

```

```

ENTITY styled_curve_occurrence
  SUBTYPE OF (managed_design_object);
    styled_curve    : curve;
    of_curve_style  : curve_style;
END_ENTITY; -- styled_curve_occurrence

```

```

ENTITY styled_text
  SUBTYPE OF (shape_item);
    text_font : font_select;
END_ENTITY; -- styled_text

```

```

ENTITY sub_assembly_relationship
  SUBTYPE OF (managed_design_object);
    related_component          : assembly_component;
    relating_component         : assembly_group_component;
    sub_assembly_reference_designation : STRING;
  UNIQUE
    UR1 : sub_assembly_reference_designation,
relating_component;
END_ENTITY; -- sub_assembly_relationship

```

```

ENTITY substitute_product
  SUBTYPE OF (managed_design_object);
    defined_alternate_product : SET[1:?] OF ee_product;
    substitutable_component   : assembly_composition_relationship;
END_ENTITY; -- substitute_product

```

```

ENTITY supplied_product_version
  SUBTYPE OF (managed_design_object);
    supplied_product_version      : ee_product_version;
    supplied_product_version_approval : ee_approval;
    supplier_part_number          : OPTIONAL ee_name;
    supplier                      : organization;
  INVERSE
    required_certification : SET[0:1] OF certification FOR certified_product_version;
  UNIQUE

```

```

        UR1 : supplier,
supplier_part_number;
END_ENTITY; -- supplied_product_version

```

```

ENTITY surface_finish_specification
  SUBTYPE OF (material_specification);
END_ENTITY; -- surface_finish_specification

```

```

ENTITY surface_prepped_terminal
  SUBTYPE OF (altered_package_terminal);
  surface_preparation : surface_finish_specification;
END_ENTITY; -- surface_prepped_terminal

```

```

ENTITY symmetrical_tolerance_characteristic
  SUBTYPE OF (tolerance_characteristic);
  SELF\coordinated_characteristic.component_characteristic : SET[2:2] OF numeric_param
END_ENTITY; -- symmetrical_tolerance_characteristic

```

```

ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- symmetry_tolerance

```

```

ENTITY tangent_plane
  SUBTYPE OF (tangent_shape_element);
END_ENTITY; -- tangent_plane

```

```

ENTITY tangent_shape_element
  SUPERTYPE OF (tangent_plane)
  SUBTYPE OF (derived_shape_element);
END_ENTITY; -- tangent_shape_element

```

```

ENTITY technology
  SUBTYPE OF (managed_design_object);
  process_technology      : process_specification;
  classification          : external_definition;
  material_product_system : ee_material;
END_ENTITY; -- technology

```

```

ENTITY template_definition
  SUBTYPE OF (ee_product_definition);
  INVERSE
    associated_template_occurrence : part_template FOR associated_definition;
END_ENTITY; -- template_definition

```

```

ENTITY template_material_cross_section_boundary
  SUBTYPE OF (non_feature_shape_element);
    associated_definition : printed_part_template_material;
  UNIQUE
    UR1 : associated_definition,
SELF\shape_element.name;
  WHERE
    WR1 : SELF\shape_element.name IN ['top',
'bottom',
'left',
'right'];
END_ENTITY; -- template_material_cross_section_boundary

```

```

ENTITY template_stratum_assignment
  SUBTYPE OF (managed_design_object);
    bound_template : template_select;
    bound_stratum : stratum;
END_ENTITY; -- template_stratum_assignment

```

```

ENTITY terminal_pair_port_equivalence_relationship
  SUBTYPE OF (managed_design_object);
    equivalent_port : analytical_model_port;
    scalar_ports : LIST[2:2] OF UNIQUE analytical_model_scalar_port;
END_ENTITY; -- terminal_pair_port_equivalence_relationship

```

```

ENTITY terminal_swap_specification
  SUBTYPE OF (change_design_object_management_relationship);
    SELF\change_design_object_management_relationship.previous_design_object : functional_design_object;
    SELF\change_design_object_management_relationship.current_design_object : functional_design_object;
END_ENTITY; -- terminal_swap_specification

```

```

ENTITY termination_constraint
  SUBTYPE OF (managed_design_object);
    constrained_termination : SET[2:?] OF mating_connector_termination;
    termination_usage_constraint : ee_requirement_occurrence;
END_ENTITY; -- termination_constraint

```

```

ENTITY tertiary_orientation_feature
  SUBTYPE OF (part_feature,
datum_feature);
    associated_body_vertical_extent : OPTIONAL SET[1:2] OF package_body_surface;
END_ENTITY; -- tertiary_orientation_feature

```

```

ENTITY test_point_part_feature
  SUBTYPE OF (part_tooling_feature);
END_ENTITY; -- test_point_part_feature

```

```

ENTITY test_requirement_allocation
  SUBTYPE OF (requirement_allocation);
  SELF\requirement_allocation.target_product_object : ee_product_version;
  SELF\requirement_allocation.allocated_requirement : test_requirement_occurrence;
  analytical_model_based_test_access                : OPTIONAL SET[1:?] OF part_feature;
  functional_specification_based_test_access         : OPTIONAL SET[1:?] OF functional_specification;
  WHERE
    WR1 : NOT (EXISTS(analytical_model_based_test_access)) OR (SIZEOF(QUERY(ta <* analyt
    WR2 : NOT (EXISTS(analytical_model_based_test_access)) OR (SIZEOF(QUERY(ta <* analyt
    WR3 : EXISTS(analytical_model_based_test_access) XOR EXISTS(functional_specification
    WR4 : NOT (EXISTS(analytical_model_based_test_access)) OR (EXISTS(SELF\requirement_a
    WR5 : NOT (EXISTS(functional_specification_based_test_access)) OR (SIZEOF(QUERY(fa <
    WR6 : NOT (EXISTS(functional_specification_based_test_access)) OR (EXISTS(SELF\requi
    WR7 : NOT (EXISTS(functional_specification_based_test_access)) OR (SIZEOF(QUERY(fa <
END_ENTITY; -- test_requirement_allocation

```

```

ENTITY test_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
  WHERE
    WR1 : SIZEOF(SELF\ee_requirement_occurrence.required_analytical_representation) + SI
END_ENTITY; -- test_requirement_occurrence

```

```

ENTITY test_select_product
  SUBTYPE OF (alternate_select_product);
END_ENTITY; -- test_select_product

```

```

ENTITY test_specification
  SUBTYPE OF (ee_specification);
END_ENTITY; -- test_specification

```

```

ENTITY theoretically_exact_dimension_representation
  SUBTYPE OF (singular_dimension_representation);
END_ENTITY; -- theoretically_exact_dimension_representation

```

```

ENTITY thermal_component
  SUBTYPE OF (assembly_component);
END_ENTITY; -- thermal_component

```

```

ENTITY thermal_feature
  SUBTYPE OF (part_feature);
END_ENTITY; -- thermal_feature

```

```

ENTITY thermal_isolation_laminate_component
  SUBTYPE OF (filled_area_material_removal_laminate_component);
  SELF\assembly_component.usage_definition : thermal_isolation_removal_template;
END_ENTITY; -- thermal_isolation_laminate_component

```

```

ENTITY thermal_isolation_removal_structure
  SUBTYPE OF (filled_area_material_removal_structure);
  INVERSE
    thermal_isolation_spacing_requirement : thermal_isolation_requirement_assignment FOR
END_ENTITY; -- thermal_isolation_removal_structure

```

```

ENTITY thermal_isolation_removal_template
  SUBTYPE OF (material_removal_feature_template);
  INVERSE
    thermal_isolation_spacing_requirement : thermal_isolation_requirement_assignment FOR
END_ENTITY; -- thermal_isolation_removal_template

```

```

ENTITY thermal_isolation_requirement_assignment
  SUBTYPE OF (managed_design_object);
  resulting_structure : thermal_isolation_removal_structure_or_temp;
  thermal_isolation_spacing_requirement : layout_spacing_requirement_occurrence;
  effective_current_capacity_requirement : ee_requirement_occurrence;
  angular_orientation_requirement : ee_requirement_occurrence;
END_ENTITY; -- thermal_isolation_requirement_assignment

```

```

ENTITY thermal_network_definition
  SUBTYPE OF (functional_unit_network_definition);
END_ENTITY; -- thermal_network_definition

```

```

ENTITY thermal_requirement_allocation
  SUBTYPE OF (assembly_requirement_allocation);
  SELF\assembly_requirement_allocation.assembly_assignment : thermal_component;
  SELF\assembly_requirement_allocation.requirement : thermal_requirement_occurrence;
END_ENTITY; -- thermal_requirement_allocation

```

```

ENTITY thermal_requirement_occurrence
  SUBTYPE OF (ee_requirement_occurrence);
END_ENTITY; -- thermal_requirement_occurrence

```



```

ENTITY thermodynamic_temperature_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- thermodynamic_temperature_data_element

```

```

ENTITY time_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- time_data_element

```

```

ENTITY tolerance_characteristic
  SUPERTYPE OF (ONEOF (minimum_tolerance_characteristic,
statistical_tolerance_characteristic,
symmetrical_tolerance_characteristic,
maximum_tolerance_characteristic,
plus_minus_tolerance_characteristic))
  SUBTYPE OF (coordinated_characteristic);
  SELF\coordinated_characteristic.component_characteristic : SET[1:?] OF numeric_param
END_ENTITY; -- tolerance_characteristic

```

```

ENTITY tolerance_specific_restraint_condition
  SUBTYPE OF (restraint_condition);
  affected_tolerances : SET[1:?] OF geometric_or_dimension_select;
END_ENTITY; -- tolerance_specific_restraint_condition

```

```

ENTITY tolerance_zone
  SUBTYPE OF (non_feature_shape_element);
  zone_boundary : tolerance_zone_boundary;
  INVERSE
    zone_definition : tolerance_zone_definition FOR defined_tolerance_zone;
END_ENTITY; -- tolerance_zone

```

```

ENTITY tolerance_zone_and_viewing_plane_orienting_relationship
  SUBTYPE OF (managed_design_object);
  orienting_viewing_plane : viewing_plane;
  oriented_tolerance_zone : tolerance_zone;
  INVERSE
    zone_orientation : viewing_plane_based_orientation FOR orienting_relationship;
END_ENTITY; -- tolerance_zone_and_viewing_plane_orienting_relationship

```

```

ENTITY tolerance_zone_boundary
  ABSTRACT SUPERTYPE OF (ONEOF (tolerance_zone_circular_or_cylindrical_or_spherical_boun
tolerance_zone_opposing_boundary_set,
conical_tolerance_zone_boundary))
  SUBTYPE OF (non_feature_shape_element);

```

```

    INVERSE
        bounded_zone          : tolerance_zone FOR zone_boundary;
        boundary_zone_definition : tolerance_zone_boundary_definition FOR defined_zone_boundar
    END_ENTITY; -- tolerance_zone_boundary

```

```

ENTITY tolerance_zone_boundary_definition
    ABSTRACT SUPERTYPE OF (ONEOF (boundary_zone_definition_with_specified_size,
conical_tolerance_zone_boundary_definition))
    SUBTYPE OF (managed_design_object);
        defined_zone_boundary : tolerance_zone_boundary;
    INVERSE
        zone_definition : tolerance_zone_definition FOR boundary_definition;
    END_ENTITY; -- tolerance_zone_boundary_definition

```

```

ENTITY tolerance_zone_boundary_member
    SUBTYPE OF (non_feature_shape_element);
    INVERSE
        explicit_opposing_boundary_set : tolerance_zone_explicit_opposing_boundary_set FOR b
    END_ENTITY; -- tolerance_zone_boundary_member

```

```

ENTITY tolerance_zone_boundary_member_offset
    SUBTYPE OF (managed_design_object);
        offset_tolerance_zone_boundary_member : tolerance_zone_boundary_member;
        offset_value                          : length_data_element;
        type_of_offset                        : offset_types;
    INVERSE
        boundary_definition : profile_boundary_definition_with_offsets FOR boundary_offsets;
    END_ENTITY; -- tolerance_zone_boundary_member_offset

```

```

ENTITY tolerance_zone_circular_or_cylindrical_or_spherical_boundary
    SUBTYPE OF (tolerance_zone_boundary);
    INVERSE
        size_characteristic : diametrical_boundary_size_characteristic FOR sized_boundary;
    END_ENTITY; -- tolerance_zone_circular_or_cylindrical_or_spherical_boundary

```

```

ENTITY tolerance_zone_definition
    SUPERTYPE OF (ONEOF (projected_tolerance_zone_definition,
runout_tolerance_zone_definition_with_specified_angle,
tolerance_zone_definition_with_per_unit_size_specification,
conical_tolerance_zone_definition,
tolerance_zone_definition_with_specified_orientation))
    SUBTYPE OF (managed_design_object);
        defined_tolerance_zone : tolerance_zone;
        boundary_definition    : tolerance_zone_boundary_definition;
        form                   : OPTIONAL tolerance_zone_form;
    END_ENTITY; -- tolerance_zone_definition

```

```

ENTITY tolerance_zone_definition_with_per_unit_size_specification
  SUBTYPE OF (tolerance_zone_definition);
    per_unit_size_specification : tolerance_zone_per_unit_size_specification;
END_ENTITY; -- tolerance_zone_definition_with_per_unit_size_specification

```

```

ENTITY tolerance_zone_definition_with_specified_orientation
  SUBTYPE OF (tolerance_zone_definition);
    zone_orientation : orientation_select;
END_ENTITY; -- tolerance_zone_definition_with_specified_orientation

```

```

ENTITY tolerance_zone_explicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_opposing_boundary_set);
    boundary_members : SET[2:2] OF tolerance_zone_boundary_member;
END_ENTITY; -- tolerance_zone_explicit_opposing_boundary_set

```

```

ENTITY tolerance_zone_form
  SUBTYPE OF (managed_design_object);
    characterized_tolerance_zone : tolerance_zone;
    name : label;
  INVERSE
    zone_definition : tolerance_zone_definition FOR form;
END_ENTITY; -- tolerance_zone_form

```

```

ENTITY tolerance_zone_implicit_opposing_boundary_set
  SUBTYPE OF (tolerance_zone_opposing_boundary_set);
END_ENTITY; -- tolerance_zone_implicit_opposing_boundary_set

```

```

ENTITY tolerance_zone_opposing_boundary_set
  ABSTRACT SUPERTYPE OF (ONEOF (tolerance_zone_explicit_opposing_boundary_set,
tolerance_zone_implicit_opposing_boundary_set))
  SUBTYPE OF (tolerance_zone_boundary);
  INVERSE
    size_characteristic : opposing_boundary_set_size_characteristic FOR sized_boundary_s
END_ENTITY; -- tolerance_zone_opposing_boundary_set

```

```

ENTITY tolerance_zone_per_unit_size_specification
  SUBTYPE OF (managed_design_object);
    specified_tolerance_zone : tolerance_zone;
    per_unit_size : SET[1:2] OF length_data_element;
  INVERSE
    zone_definition : tolerance_zone_definition_with_per_unit_size_specification FOR per
END_ENTITY; -- tolerance_zone_per_unit_size_specification

```

```

ENTITY topological_junction
  SUBTYPE OF (managed_design_object);
    scope : physical_connectivity_structure_definition;
    name   : ee_name;
  UNIQUE
    UR1 : name,
scope;
END_ENTITY; -- topological_junction

```

```

ENTITY total_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_referenced_datum_system);
END_ENTITY; -- total_runout_tolerance

```

```

ENTITY trace_template
  SUPERTYPE OF (default_trace_template)
  SUBTYPE OF (part_template);
    trace_style : curve_style;
END_ENTITY; -- trace_template

```

```

ENTITY transmission_line_functional_unit_terminal_allocation
  SUBTYPE OF (managed_design_object);
    implementation                : printed_component_join_terminal;
    transmission_line_functional_unit_terminal : functional_unit_terminal;
    functional_unit_design_path    : transmission_line_functional_unit_to_routed_printed_component_allocation;
  UNIQUE
    UR1 : implementation,
transmission_line_functional_unit_terminal;
END_ENTITY; -- transmission_line_functional_unit_terminal_allocation

```

```

ENTITY transmission_line_functional_unit_to_routed_printed_component_allocation
  SUBTYPE OF (design_functional_unit_allocation_to_assembly_component);
    SELF\design_functional_unit_allocation_to_assembly_component.implementation : routed_printed_component_allocation;
END_ENTITY; -- transmission_line_functional_unit_to_routed_printed_component_allocation

```

```

ENTITY trimmed_line
  SUBTYPE OF (curve);
    start_point : cartesian_point;
    end_point   : cartesian_point;
END_ENTITY; -- trimmed_line

```

```

ENTITY true_profile
  SUBTYPE OF (managed_design_object);
    profiled_surface : physical_feature_or_part_template;

```

```

    profile_representation : true_profile_representations;
INVERSE
    boundary_definition : SET[1:?] OF profile_boundary_definition FOR basic_profile;
END_ENTITY; -- true_profile

```

```

ENTITY type_qualifier
    SUBTYPE OF (managed_design_object);
    name : label;
END_ENTITY; -- type_qualifier

```

```

ENTITY uncertainty_qualifier
    SUPERTYPE OF (ONEOF (standard_uncertainty,
qualitative_uncertainty))
    SUBTYPE OF (managed_design_object);
    measure_name : text;
    description : text;
END_ENTITY; -- uncertainty_qualifier

```

```

ENTITY unrouted_conductive_interconnect_element
    SUBTYPE OF (conductive_interconnect_element);
    connected_points : SET[1:?] OF layer_connection_point_link;
    WHERE
        WR1 : NOT (EXISTS(SELF\conductive_interconnect_element.composed_conductor));
END_ENTITY; -- unrouted_conductive_interconnect_element

```

```

ENTITY unrouted_join_relationship
    SUBTYPE OF (join_relationship);
    unrouted_terminals : SET[2:?] OF laminate_component_join_terminal;
END_ENTITY; -- unrouted_join_relationship

```

```

ENTITY unsupported_passage
    SUBTYPE OF (inter_stratum_feature);
    SELF\assembly_component.usage_definition : unsupported_passage_template;
END_ENTITY; -- unsupported_passage

```

```

ENTITY unsupported_passage_dependent_non_functional_land
    SUBTYPE OF (non_functional_land);
    reference_passage : unsupported_passage;
    SELF\assembly_component.usage_definition : default_unsupported_passage_based_land_ph
END_ENTITY; -- unsupported_passage_dependent_non_functional_land

```

```

ENTITY unsupported_passage_template

```

```

    SUPERTYPE OF (ONEOF (counterbore_passage_template,
countersunk_passage_template,
blind_passage_template))

```

```

    SUBTYPE OF (inter_stratum_feature_template);
    SELF\inter_stratum_feature_template.of_passage_technology : default_unsupported_passage_technology;
END_ENTITY; -- unsupported_passage_template

```

```

ENTITY usage_concept_usage_relationship
  SUBTYPE OF (managed_design_object);
    associating_non_feature_usage_shape : OPTIONAL non_feature_shape_definition;
    associating_usage_shape              : OPTIONAL part_template_or_physical_unit_shape_definition;
    associated_usage                     : usage_concept;
    associated_shape_definition          : shape_definition;
    associated_usage_placement           : axis_placement_or_cto2d;
  WHERE
    WR1 : (EXISTS(associating_non_feature_usage_shape)) XOR (EXISTS(associating_usage_shape));
    WR2 : NOT (EXISTS(associating_non_feature_usage_shape)) OR ('AP210_ARM.NON_FEATURE_SHAPE_DEFINITION' IN TYPEOF(associated_usage_concept));
    WR3 : NOT (EXISTS(associating_usage_shape)) OR ('AP210_ARM.FEATURE_SHAPE_DEFINITION' IN TYPEOF(associated_shape_definition));
    WR4 : NOT ('AP210_ARM.CARTESIAN_TRANSFORMATION_OPERATOR_2D' IN TYPEOF(associated_usage_placement));
END_ENTITY; -- usage_concept_usage_relationship

```

```

ENTITY usage_view_connection_zone_terminal_shape_relationship
  SUBTYPE OF (managed_design_object);
    associating_terminal_shape           : feature_shape_definition;
    associated_usage                     : connection_zone;
    associated_connection_zone_shape_definition : feature_shape_definition;
    associated_usage_placement           : axis_placement;
  WHERE
    WR1 : associating_terminal_shape :<> associated_connection_zone_shape_definition;
END_ENTITY; -- usage_view_connection_zone_terminal_shape_relationship

```

```

ENTITY user_defined_dimension_qualifier
  SUBTYPE OF (dimension_qualifier);
    description : text;
    name        : text;
END_ENTITY; -- user_defined_dimension_qualifier

```

```

ENTITY user_defined_geometric_tolerance_qualifier
  SUBTYPE OF (geometric_tolerance_qualifier);
    description : text;
    name        : text;
END_ENTITY; -- user_defined_geometric_tolerance_qualifier

```

```

ENTITY vector
  SUBTYPE OF (managed_design_object);
END_ENTITY; -- vector

```

```

ENTITY via
  ABSTRACT SUPERTYPE OF (ONEOF (blind_via,
buried_via,
interfacial_connection))
  SUBTYPE OF (plated_passage);
  SELF\assembly_component.usage_definition : via_template;
END_ENTITY; -- via

```

```

ENTITY via_and_contact_size_dependent_land
  SUBTYPE OF (land);
  SELF\assembly_component.usage_definition : default_attachment_size_and_via_based_land;
  reference_via : via;
  INVERSE
    interface_access_mechanism : land_interface_terminal FOR associated_component;
END_ENTITY; -- via_and_contact_size_dependent_land

```

```

ENTITY via_and_contact_size_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  reference_via : via;
  SELF\assembly_component.usage_definition : default_attachment_size_and_via_based_land;
  INVERSE
    interface_access_mechanism : non_functional_land_interface_terminal FOR associated_component;
END_ENTITY; -- via_and_contact_size_dependent_non_functional_land

```

```

ENTITY via_dependent_land
  SUBTYPE OF (land);
  reference_via : via;
  SELF\assembly_component.usage_definition : default_via_based_land_physical_template;
END_ENTITY; -- via_dependent_land

```

```

ENTITY via_dependent_non_functional_land
  SUBTYPE OF (non_functional_land);
  reference_via : via;
  SELF\assembly_component.usage_definition : default_via_based_land_physical_template;
END_ENTITY; -- via_dependent_non_functional_land

```

```

ENTITY via_template
  SUBTYPE OF (inter_stratum_feature_template);
  SELF\inter_stratum_feature_template.of_passage_technology : default_via_definition;
  INVERSE
    access_mechanisms : SET[2:?] OF via_template_terminal FOR associated_definition;
END_ENTITY; -- via_template

```

```

ENTITY via_template_terminal
  SUBTYPE OF (shape_element);
    associated_definition          : via_template;
    disallowed_inter_stratum_extent : OPTIONAL inter_stratum_extent;
    SELF\shape_element.name       : ee_name;
    terminal_connection_zone       : OPTIONAL SET[1:?] OF connection_zone;
  INVERSE
    terminal_shape : SET[0:1] OF usage_concept_usage_relationship FOR associated_usage;
  UNIQUE
    UR1 : name,
associated_definition;
END_ENTITY; -- via_template_terminal

```

```

ENTITY via_terminal
  SUBTYPE OF (laminate_component_feature);
    SELF\component_feature.definition          : via_template_terminal;
    SELF\component_feature.associated_component : via;
END_ENTITY; -- via_terminal

```

```

ENTITY view_level_non_feature_shape_definition
  SUBTYPE OF (non_feature_shape_definition);
    SELF\non_feature_shape_definition.model_shape : view_shape_select;
END_ENTITY; -- view_level_non_feature_shape_definition

```

```

ENTITY viewing_plane
  SUBTYPE OF (non_feature_shape_element);
  INVERSE
    viewing_plane_specification : viewing_plane_definition FOR defined_viewing_plane;
END_ENTITY; -- viewing_plane

```

```

ENTITY viewing_plane_based_orientation
  SUBTYPE OF (managed_design_object);
    orienting_relationship : tolerance_zone_and_viewing_plane_orienting_relationship;
    viewing_plane_specification : viewing_plane_definition;
END_ENTITY; -- viewing_plane_based_orientation

```

```

ENTITY viewing_plane_definition
  SUBTYPE OF (managed_design_object);
    defined_viewing_plane : viewing_plane;
    viewing_plane_representation : axis_placement;
  INVERSE
    zone_orientation : viewing_plane_based_orientation FOR viewing_plane_specification;
END_ENTITY; -- viewing_plane_definition

```



```

ENTITY volume_data_element
  SUBTYPE OF (data_element);
END_ENTITY; -- volume_data_element

```

```

ENTITY volume_shape_intersection
  SUPERTYPE OF (ONEOF (edge_segment_cross_section,
pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship,
pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship,
pre_defined_parallel_datum_axis_symbol_3d_2d_relationship))
  SUBTYPE OF (managed_design_object);
  surface_intersection_result      : planar_shape;
  projected_volume                : bound_volume_shape;
  orientation_intersection_plane  : axis_placement_3d;
  planar_shape_origin            : axis_placement_2d;
END_ENTITY; -- volume_shape_intersection

```

```

ENTITY volume_shape_plane_projection
  SUBTYPE OF (managed_design_object);
  surface_projection_result : planar_shape;
  projected_volume         : bound_volume_shape;
  orientation_viewing_plane : axis_placement_3d;
  viewing_plane            : axis_placement_3d;
  planar_shape_origin      : axis_placement_2d;
END_ENTITY; -- volume_shape_plane_projection

```

```

ENTITY volume_shape_point_projection
  SUBTYPE OF (managed_design_object);
  surface_projection_result : planar_shape;
  projected_volume         : bound_volume_shape;
  orientation_viewing_plane : axis_placement_3d;
  viewing_point            : cartesian_point;
  planar_shape_origin      : axis_placement_2d;
END_ENTITY; -- volume_shape_point_projection

```

```

ENTITY wire_terminal
  SUBTYPE OF (package_terminal);
  SELF\package_terminal.internal_connection_zone : SET[1:?] OF connection_zone;
  maximum_wire_terminal_length                  : length_data_element;
  minimum_wire_terminal_length                  : length_data_element;
END_ENTITY; -- wire_terminal

```

```

ENTITY work_order
  ABSTRACT SUPERTYPE OF (ONEOF (change_order,
start_order))
  SUBTYPE OF (managed_design_object);

```

```

referenced_work_request : SET[1:?] OF work_request;
work_order_id           : ee_name;
resulting_product       : SET[1:?] OF ee_product_version;
analysis_data           : OPTIONAL ee_text;
additional_data         : OPTIONAL ee_text;
work_order_approval     : ee_approval;
initiator               : SET[1:?] OF person_organization;
start_date              : date_or_date_and_time;
release_date            : date_or_date_and_time;
authorizing_agreement   : OPTIONAL contract;
UNIQUE
  UR1 : work_order_id;
END_ENTITY; -- work_order

```

```

ENTITY work_request
  ABSTRACT SUPERTYPE OF (ONEOF (start_request,
change_request))
  SUBTYPE OF (managed_design_object);
  work_request_approval : ee_approval;
  request_recipient     : SET[1:?] OF person_organization;
  work_request_id       : ee_name;
  referenced_product    : SET[1:?] OF ee_product_version;
  request_date          : date_or_date_and_time;
  description           : ee_text;
  reason                : evaluated_characteristic_or_description;
  initiator             : SET[1:?] OF person_organization;
  status                : STRING;
  INVERSE
    implementation_order : SET[0:1] OF work_order FOR referenced_work_request;
  UNIQUE
    UR1 : work_request_id;
END_ENTITY; -- work_request

```

```

FUNCTION assigned_product_category_correlation(
  input1 : SET OF ee_product_version ;
  input2 : SET OF ee_product_related_product_category
) : BOOLEAN;

  LOCAL
    d_role_bag : BAG OF STRING;
  END_LOCAL;

  REPEAT i := 1 TO SIZEOF(input2) BY 1;
    d_role_bag := input2[i]\ee_product_category.name + d_role_bag;
  END_REPEAT;
  REPEAT i := 1 TO SIZEOF(d_role_bag) BY 1;
    CASE d_role_bag[i] OF
      'assembly'           : RETURN(TRUE);
      'bare die'           : RETURN(TRUE);
      'functionality'      : RETURN(TRUE);
    END CASE;
  END_REPEAT;

```

```

        'interconnect'           : RETURN(TRUE);
        'material'               : RETURN(TRUE);
        'package'                : RETURN(TRUE);
        'physical network design' : RETURN(TRUE);
        'piece part'             : RETURN(TRUE);
        'printed part'           : RETURN(TRUE);
        'requirements model'      : RETURN(TRUE);
        'rule'                   : RETURN(TRUE);
        'simulation model'        : RETURN(TRUE);
        'standard part'           : RETURN(TRUE);
        'technology specific model' : RETURN(TRUE);
        'template model'         : RETURN(TRUE);
        OTHERWISE                 : ;
    END_CASE;
END_REPEAT;
RETURN(FALSE);
END_FUNCTION; -- assigned_product_category_correlation

```

```

FUNCTION add_data_element(
    input1 : length_data_element ;
    input2 : length_data_element
) : length_data_element;

    RETURN(input1);
END_FUNCTION; -- add_data_element

```

```

FUNCTION consistent_bus_structural_definition(
    input : SET OF bus_element_link
) : BOOLEAN;

    LOCAL
        vertex      : SET OF managed_design_object := [];
        edge        : INTEGER := SIZEOF(input);
        vc          : INTEGER := 0;
        vertex_degree : INTEGER := 0;
    END_LOCAL;

    REPEAT i := 1 TO SIZEOF(input) BY 1;
        vertex := vertex + input[i].precedent_element;
        vertex := vertex + input[i].subsequent_element;
    END_REPEAT;
    vc := SIZEOF(vertex);
    IF (vc <> (edge + 1)) THEN
        RETURN(FALSE);
    END_IF;
    REPEAT i := 1 TO vc BY 1;
        vertex_degree := 0;
        REPEAT j := 1 TO SIZEOF(input) BY 1;
            IF input[j].precedent_element = vertex[i] THEN
                vertex_degree := vertex_degree + 1;
            END_IF;
        END_REPEAT;
    END_REPEAT;

```

```

        END_IF;
        IF input[j].subsequent_element = vertex[i] THEN
            vertex_degree := vertex_degree + 1;
        END_IF;
    END_REPEAT;
    IF (vertex_degree > 2) THEN
        RETURN(FALSE);
    END_IF;
    END_REPEAT;
    RETURN(TRUE);
END_FUNCTION; -- consistent_bus_structural_definition

```

```

FUNCTION consistent_stack_definition(
    input1 : printed_part_template ;
    input2 : SET OF printed_component_link
) : BOOLEAN;

    LOCAL
        pass : BOOLEAN := FALSE;
    END_LOCAL;

    REPEAT i := 1 TO SIZEOF(input2) BY 1;
        pass := input2[i].definition.associated_printed_part_template :=: input1;
    END_REPEAT;
    RETURN(pass);
END_FUNCTION; -- consistent_stack_definition

```

```

FUNCTION correlated_terminals(
    input1 : SET OF conductive_interconnect_element_terminal ;
    input2 : SET OF conductive_interconnect_element_terminal_link
) : BOOLEAN;

    LOCAL
        pass : BOOLEAN := FALSE;
        ciet : SET OF conductive_interconnect_element_terminal := [];
    END_LOCAL;

    REPEAT i := 1 TO SIZEOF(input2) BY 1;
        ciet := ciet + input2[i].precedent_point;
        ciet := ciet + input2[i].subsequent_point;
    END_REPEAT;
    pass := (input1 = ciet);
    RETURN(pass);
END_FUNCTION; -- correlated_terminals

```

```

FUNCTION derive_mirror(
    cto2d : cartesian_transformation_operator_2d
) : BOOLEAN;

```

```

    RETURN(TRUE);
END_FUNCTION; -- derive_mirror

```

```

FUNCTION disjoint_package_terminals(
    input : packaged_part_terminal
) : BOOLEAN;

    RETURN(TRUE);
END_FUNCTION; -- disjoint_package_terminals

```

```

FUNCTION get_cit(
    ac : assembly_component ;
    ajb : SET OF assembly_joint
) : component_interface_terminal;

    LOCAL
        cf : BAG OF component_feature := [];
        cit : component_interface_terminal;
    END_LOCAL;

    cf[1] := ajb[1].assembly_features_1;
    cf[2] := ajb[1].assembly_features_2;
    REPEAT i := 1 TO 2 BY 1;
        IF ac := cf[i].associated_component THEN
            IF ('AP210_ARM.INTERFACE_COMPONENT' IN TYPEOF(cf[i].associated_component)) THEN
                RETURN(cf[i]);
            ELSE
                RETURN(?);
            END_IF;
        ELSE
            RETURN(?);
        END_IF;
    END_REPEAT;
END_FUNCTION; -- get_cit

```

```

FUNCTION get_pctd(
    ac : assembly_component ;
    ajb : SET OF assembly_joint
) : SET OF part_connected_terminals_definition;

    LOCAL
        cf : BAG OF component_feature := [];
        pctd : SET OF part_connected_terminals_definition := [];
    END_LOCAL;

    cf[1] := ajb[1].assembly_features_1;
    cf[2] := ajb[1].assembly_features_2;

```

```

REPEAT i := 1 TO 2 BY 1;
  IF ac :=: cf[i].associated_component THEN
    IF ('AP210_ARM.PART_TERMINAL' IN TYPEOF(cf[i].definition)) THEN
      pctd[1] := cf[i].definition.connection_requirement;
      RETURN(pctd);
    ELSE
      RETURN(pctd);
    END_IF;
  ELSE
    RETURN(pctd);
  END_IF;
END_REPEAT;
END_FUNCTION; -- get_pctd

```

```

FUNCTION get_pctr(
  ac  : assembly_component ;
  imjb : SET OF interface_mounted_join
) : SET OF packaged_connector_terminal_relationship;

  LOCAL
    cf  : component_feature;
    pctr : SET OF packaged_connector_terminal_relationship := [];
  END_LOCAL;

  cf := imjb[1].connection_to_interconnect_module_component;
  IF ac :=: cf.associated_component THEN
    IF ('AP210_ARM.PACKAGED_PART_INTERFACE_TERMINAL' IN TYPEOF(cf.definition)) THEN
      pctr[1] := cf.definition.external_connector_function;
      RETURN(pctr);
    ELSE
      RETURN(pctr);
    END_IF;
  ELSE
    RETURN(pctr);
  END_IF;
END_FUNCTION; -- get_pctr

```

```

FUNCTION max_data_element(
  input1 : length_data_element ;
  input2 : length_data_element
) : length_data_element;

  RETURN(input1);
END_FUNCTION; -- max_data_element

```

```

FUNCTION single_package_version(
  input : packaged_part
) : BOOLEAN;

```

```

    RETURN(TRUE);
END_FUNCTION; -- single_package_version

```

```

FUNCTION substrate_in_assembly(
    c2dl : component_2d_location
) : BOOLEAN;

```

```

    RETURN(TRUE);
END_FUNCTION; -- substrate_in_assembly

```

```

FUNCTION printed_component_access_mechanisms(
    input : SET OF printed_component_link
) : SET OF printed_component_join_terminal;

```

```

    LOCAL
        pcjt : SET OF printed_component_join_terminal := [];
    END_LOCAL;

```

```

    REPEAT i := 1 TO SIZEOF(input) BY 1;
        pcjt := input[i].precedent_printed_component.raw_access_mechanism + pcjt;
        pcjt := input[i].subsequent_printed_component.raw_access_mechanism + pcjt;
    END_REPEAT;
    RETURN(pcjt);
END_FUNCTION; -- printed_component_access_mechanisms

```

```

FUNCTION printed_part_template_access_mechanisms(
    input : SET OF printed_part_template_link
) : SET OF printed_part_template_terminal;

```

```

    LOCAL
        pptt : SET OF printed_part_template_terminal := [];
    END_LOCAL;

```

```

    REPEAT i := 1 TO SIZEOF(input) BY 1;
        pptt := input[i].precedent_printed_part_template.raw_access_mechanisms + pptt;
        pptt := input[i].subsequent_printed_part_template.raw_access_mechanisms + pptt;
    END_REPEAT;
    RETURN(pptt);
END_FUNCTION; -- printed_part_template_access_mechanisms

```

```

FUNCTION same_stratum(
    input : SET OF stratum_feature
) : BOOLEAN;

```

```

    LOCAL
        pass : BOOLEAN := FALSE;

```

```
s      : SET OF stratum := [];  
pptt  : SET OF printed_part_template_terminal := [];  
END_LOCAL;  
  
REPEAT i := 1 TO SIZEOF(input) BY 1;  
  s := input[i].resident_stratum + s;  
END_REPEAT;  
pass := (SIZEOF(s) = 1);  
RETURN(pass);  
END_FUNCTION; -- same_stratum  
  
FUNCTION valid_constraining_shape(  
  sd   : SET OF shape_definition ;  
  eepd : ee_product_definition  
) : BOOLEAN;  
  
  RETURN(TRUE);  
END_FUNCTION; -- valid_constraining_shape  
  
END_SCHEMA; -- AP210_ARM
```

H.2 EXPRESS-G

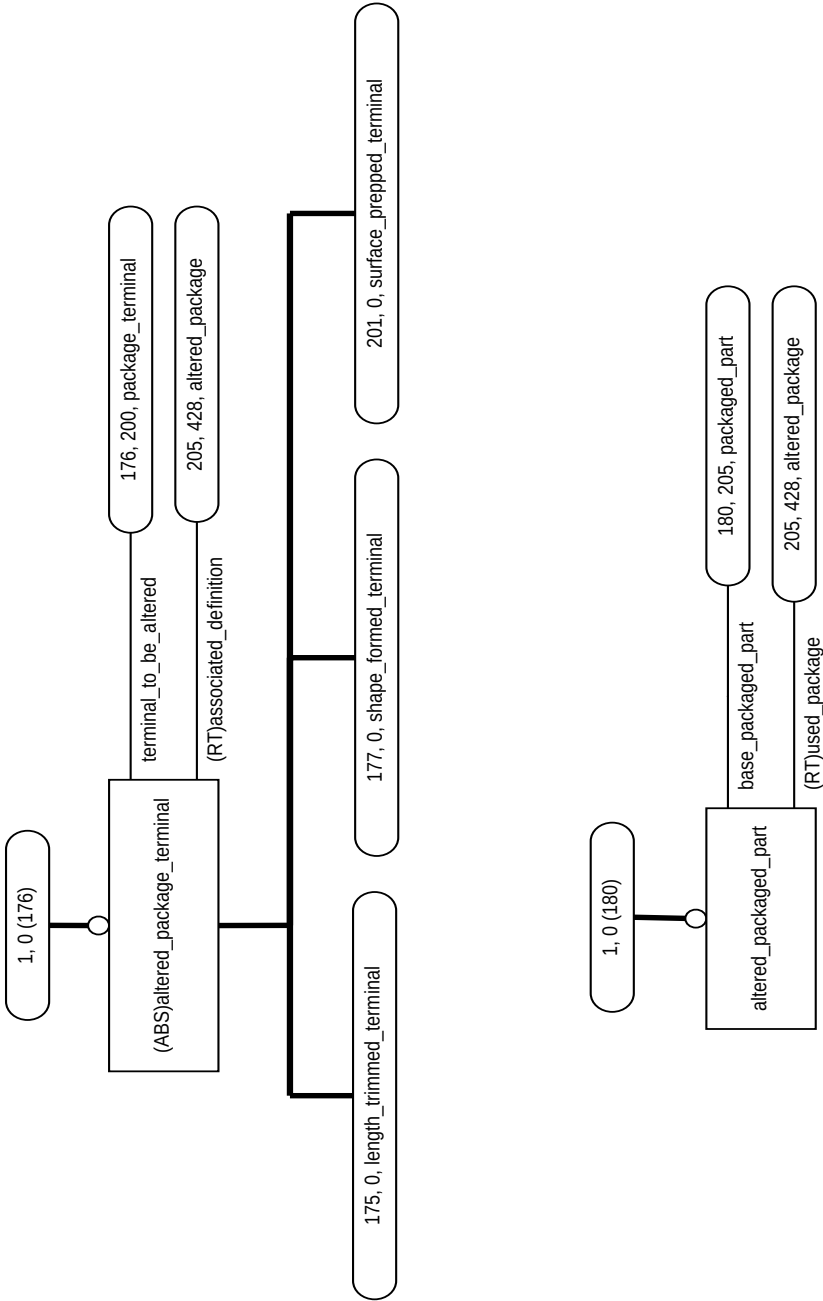


Figure H.1 – ARM EXPRESS-G Graphic 1 of 243

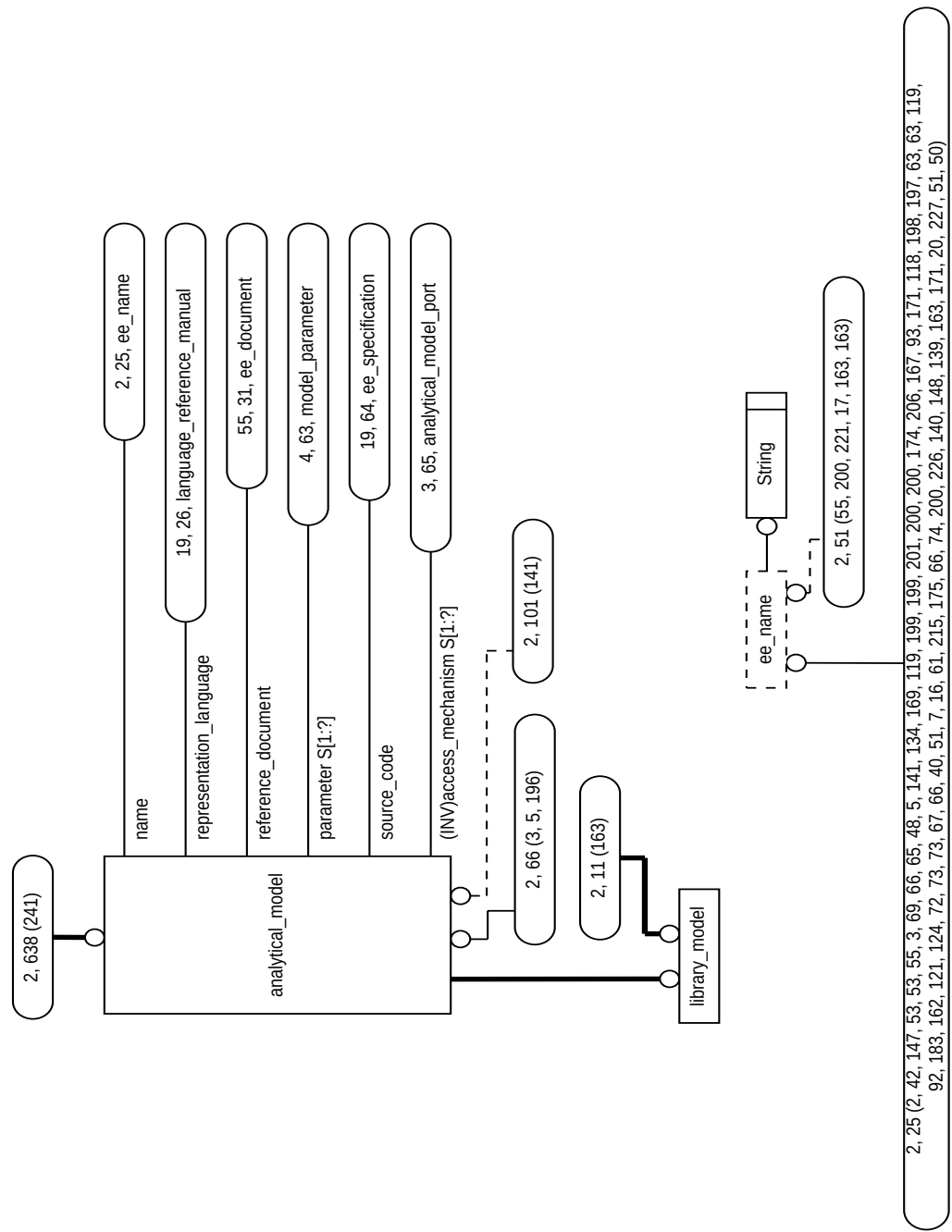


Figure H.2 – ARM EXPRESS-G Graphic 2 of 243

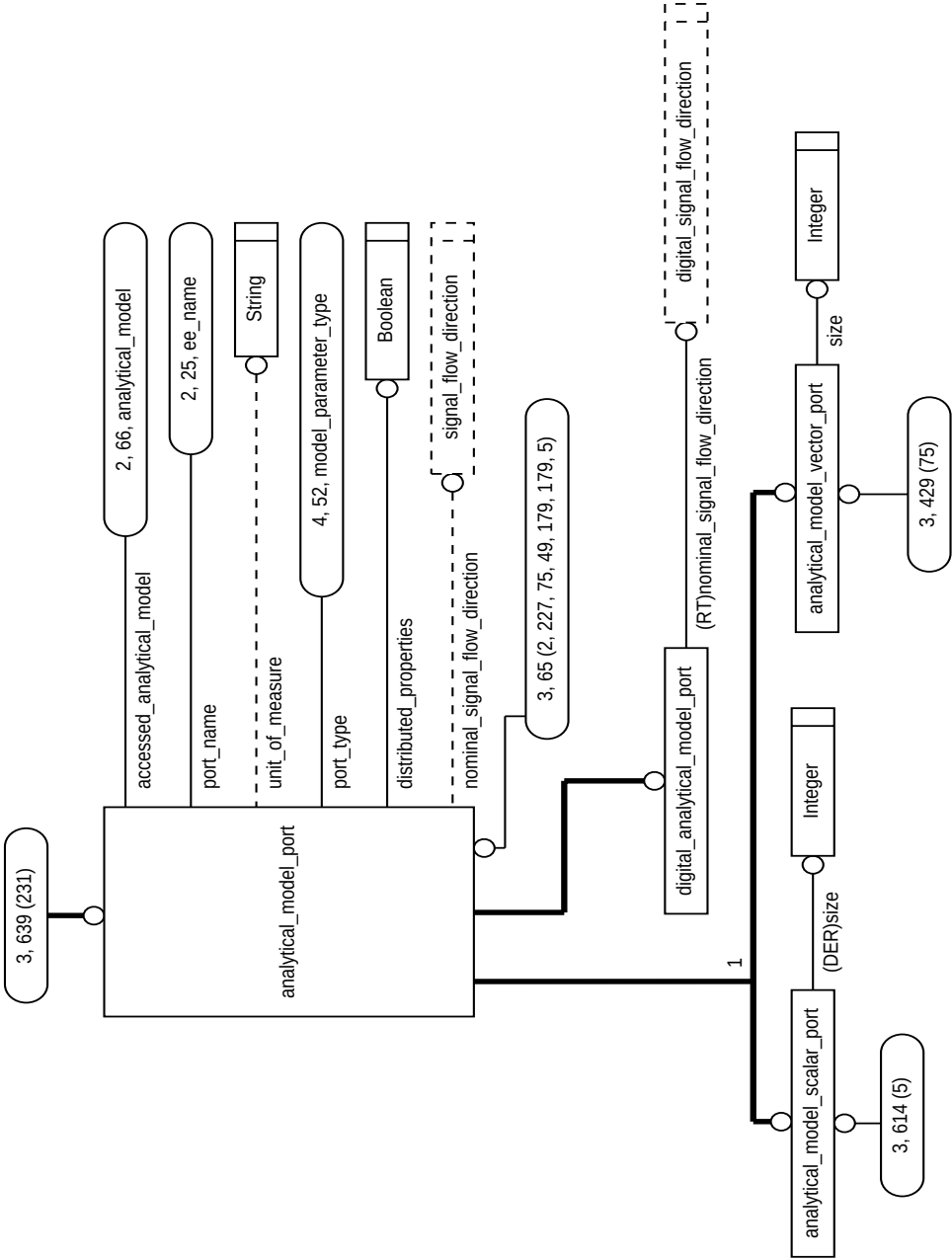


Figure H.3 – ARM EXPRESS-G Graphic 3 of 243

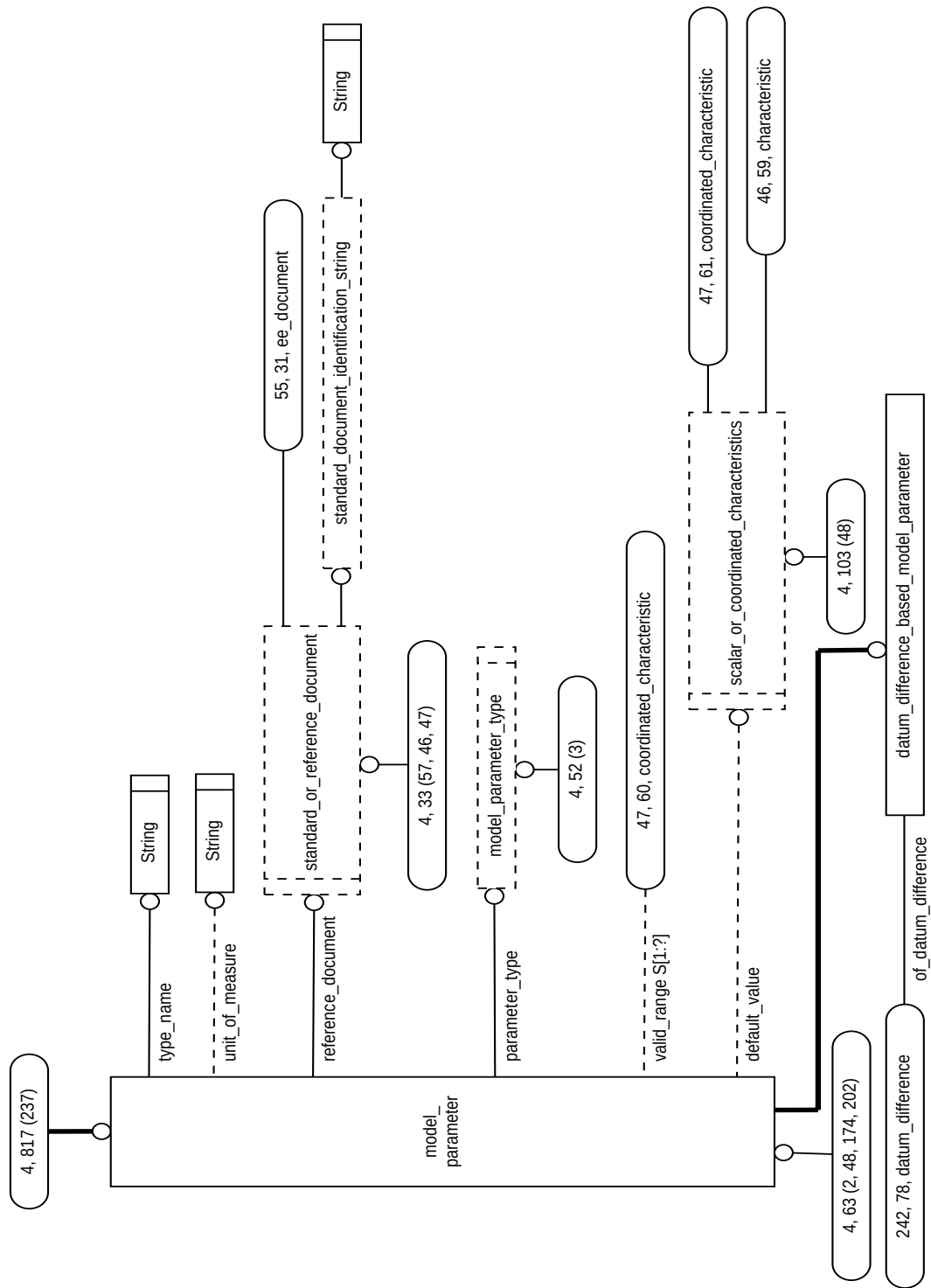


Figure H.4 – ARM EXPRESS-G Graphic 4 of 243

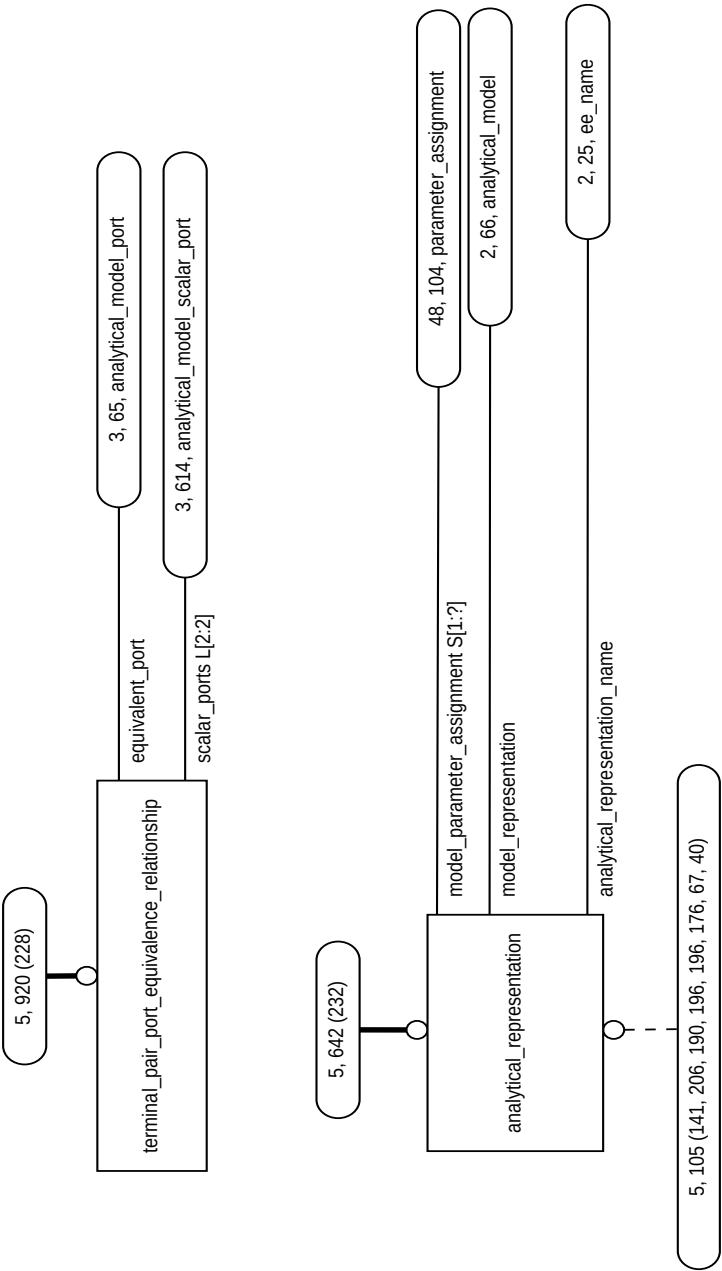


Figure H.5 – ARM EXPRESS-G Graphic 5 of 243

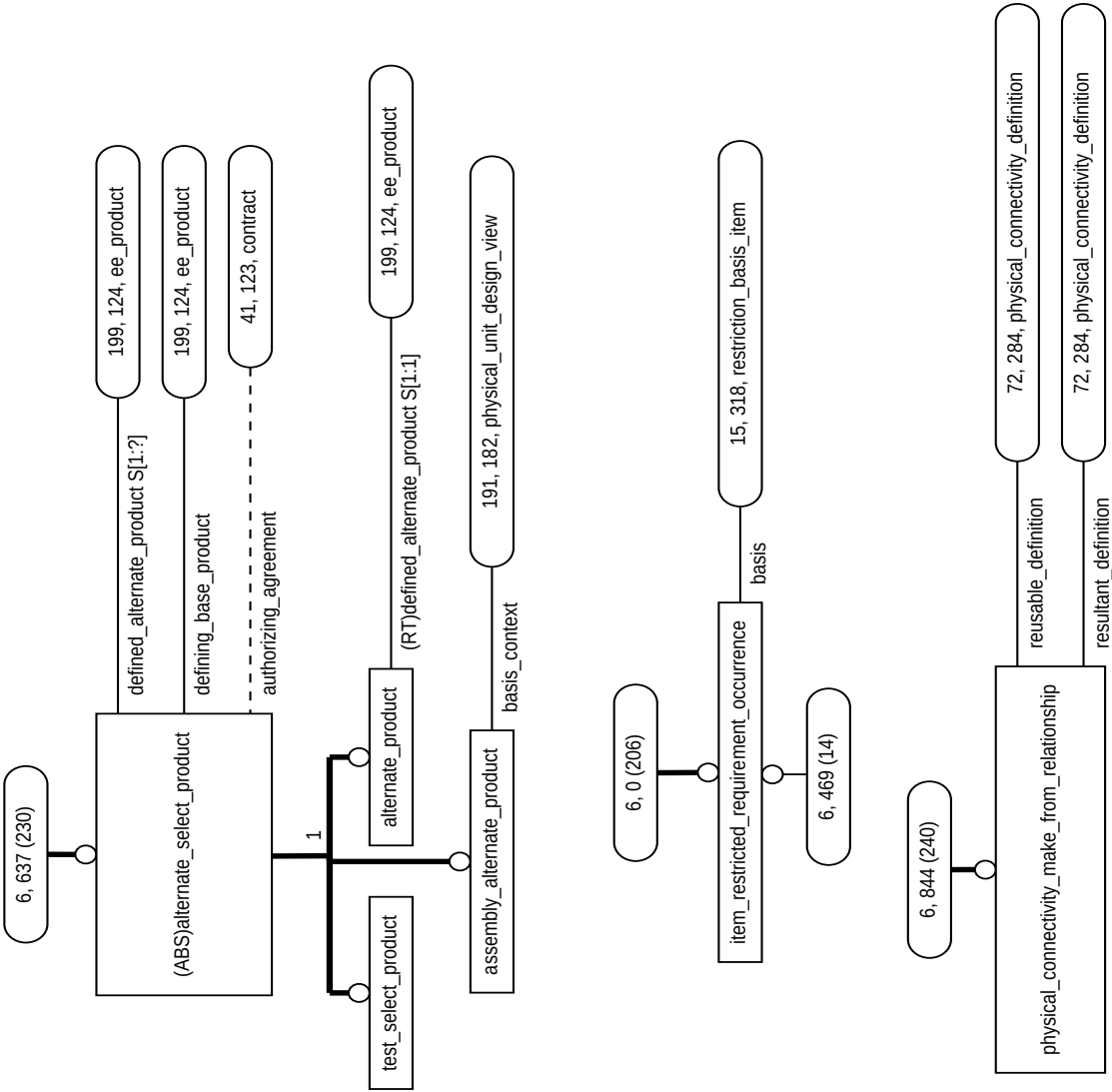


Figure H.6 – ARM EXPRESS-G Graphic 6 of 243

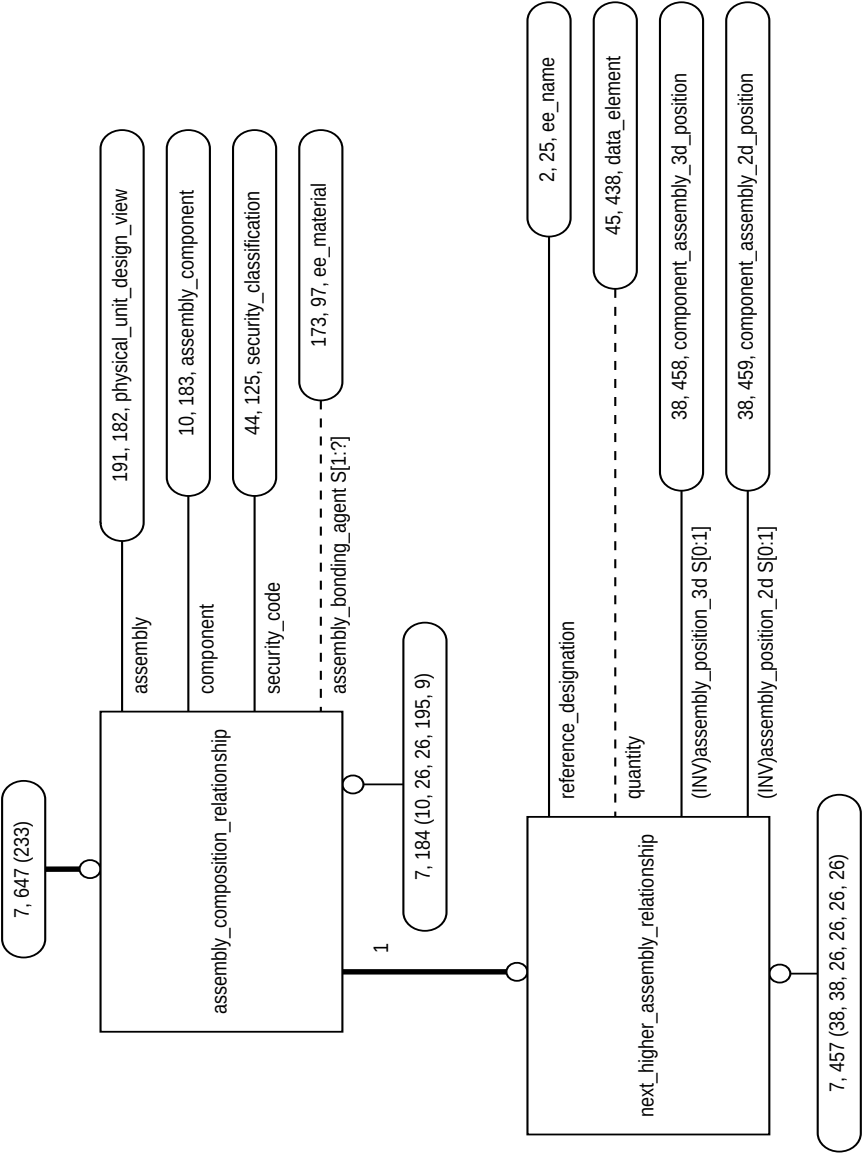


Figure H.7 – ARM EXPRESS-G Graphic 7 of 243

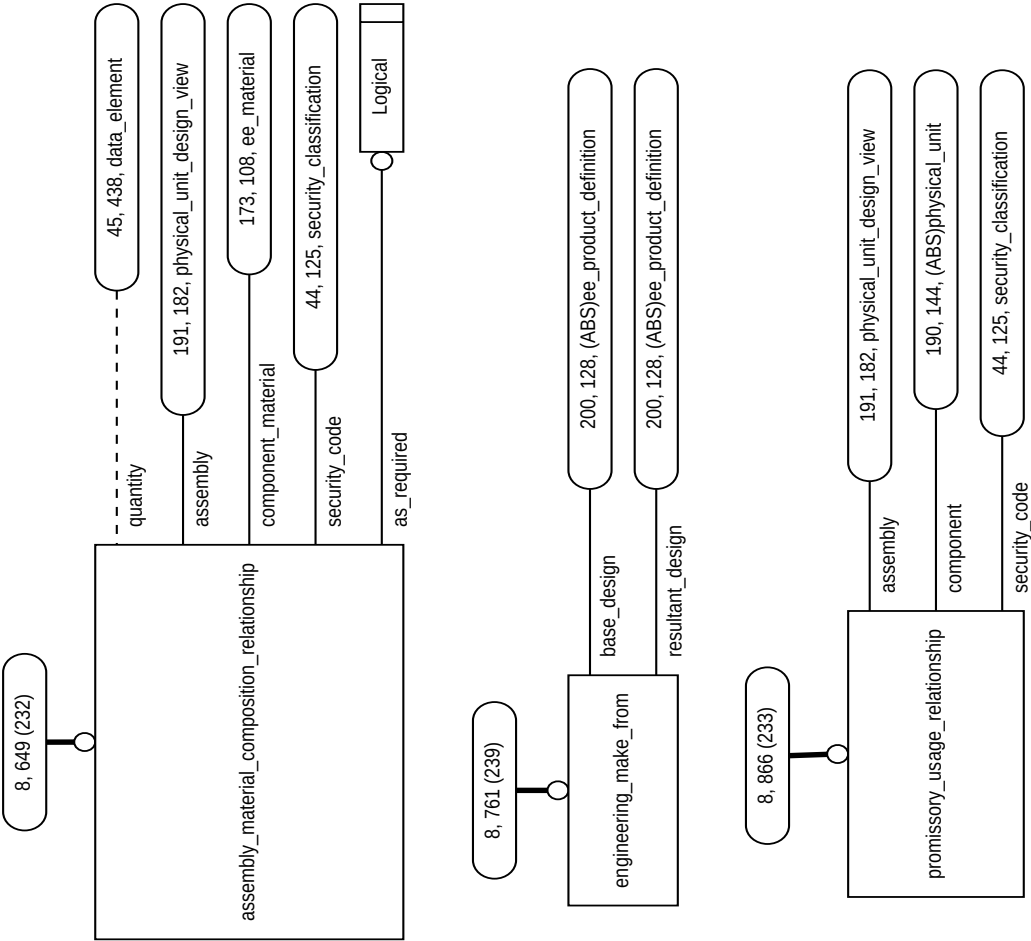


Figure H.8 – ARM EXPRESS-G Graphic 8 of 243

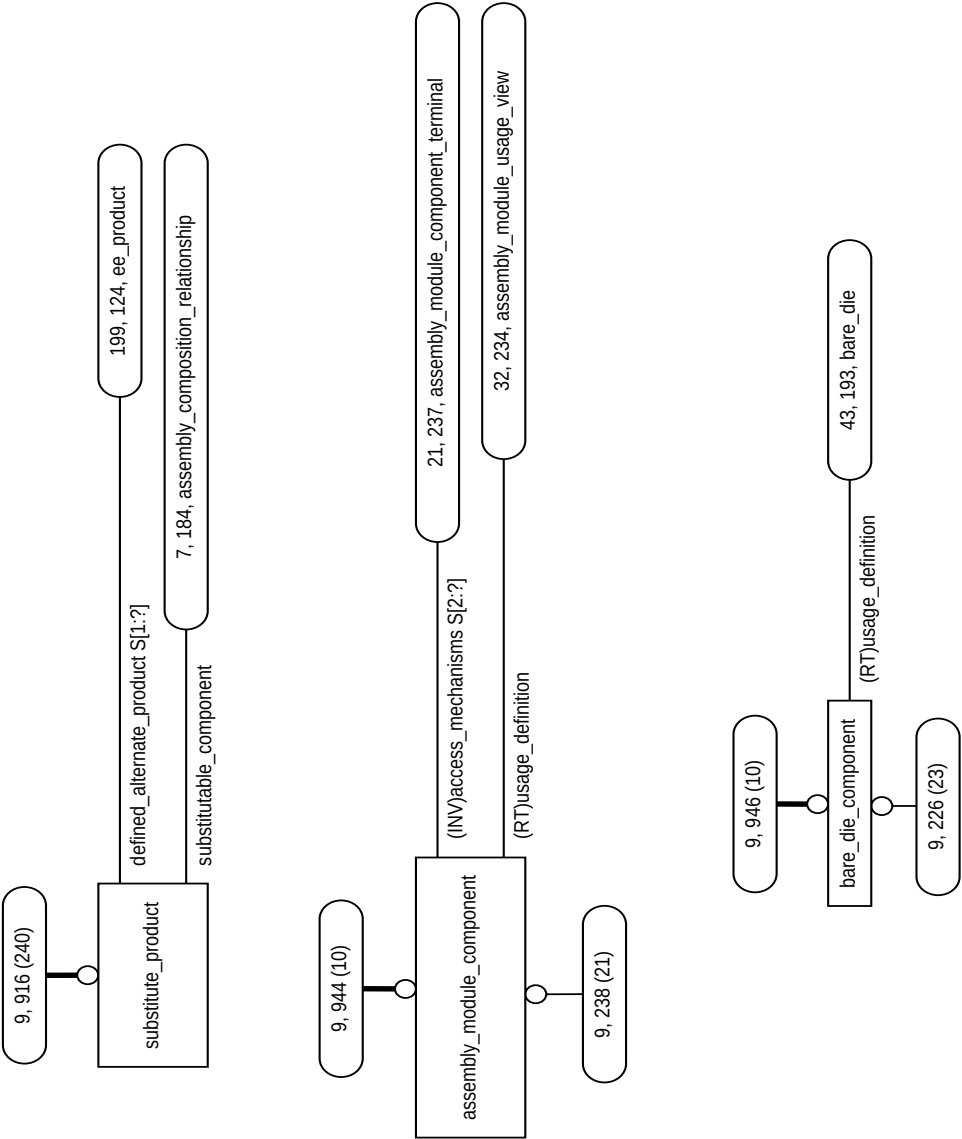


Figure H.9 – ARM EXPRESS-G Graphic 9 of 243

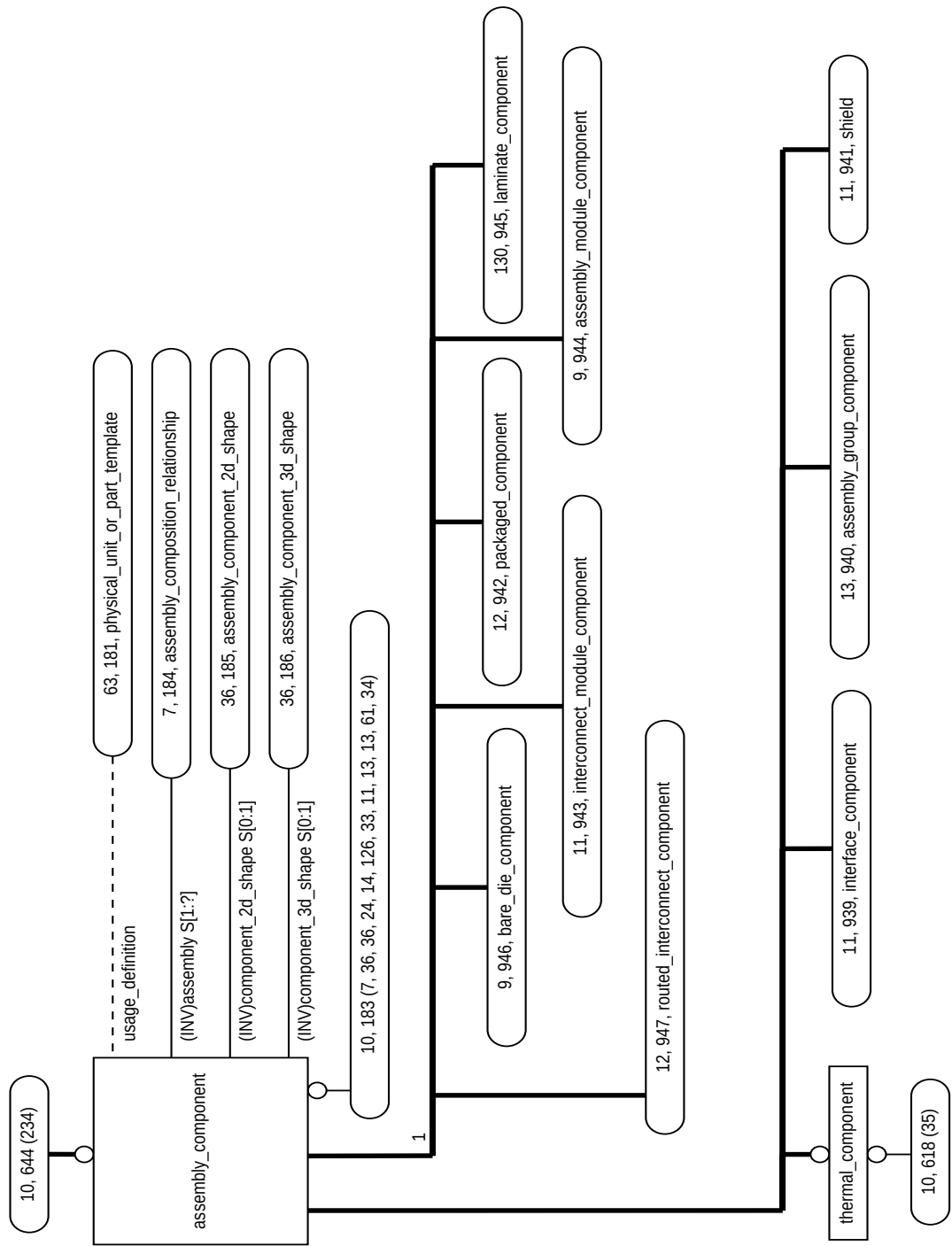


Figure H.10 – ARM EXPRESS-G Graphic 10 of 243

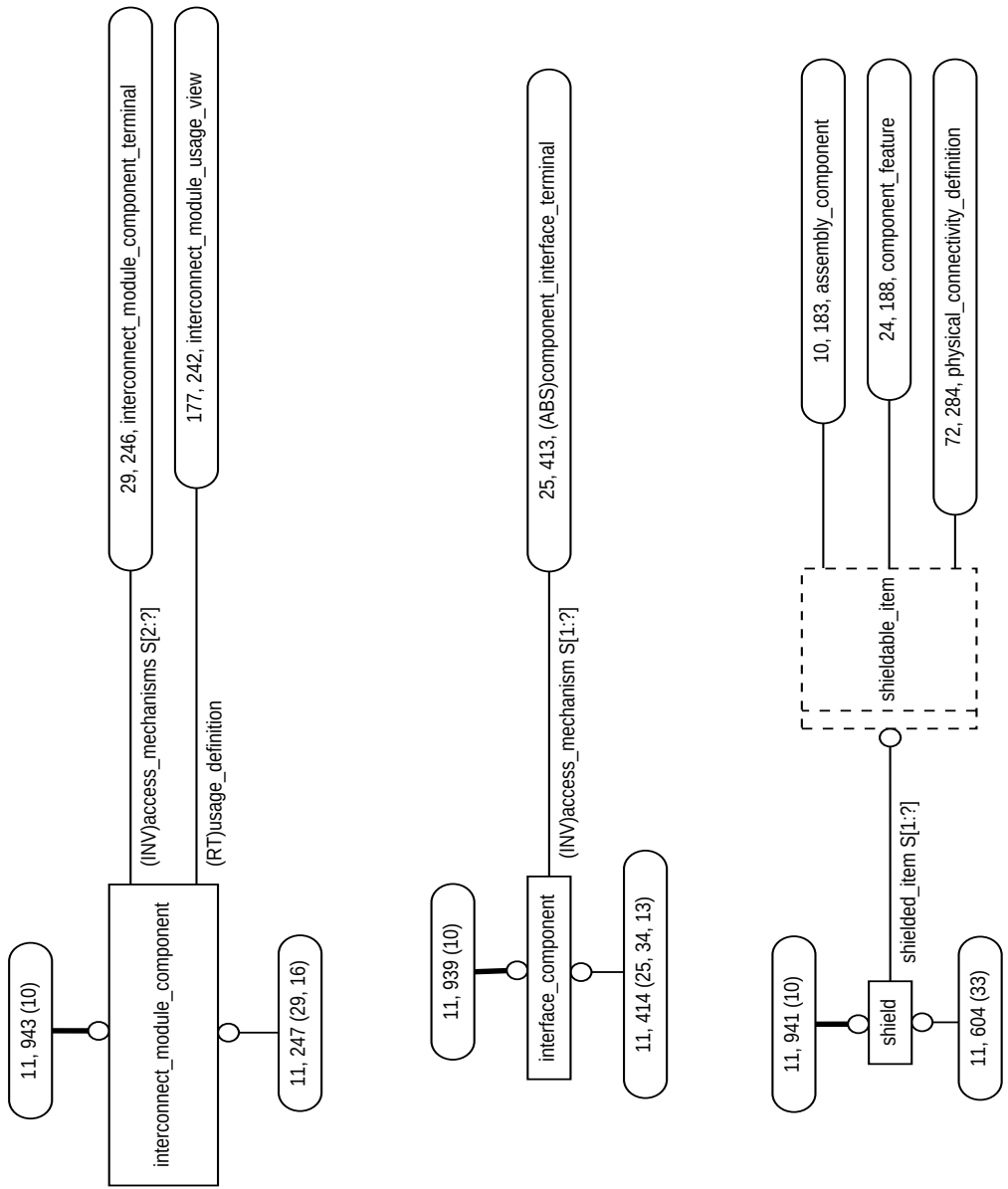


Figure H.11 – ARM EXPRESS-G Graphic 11 of 243

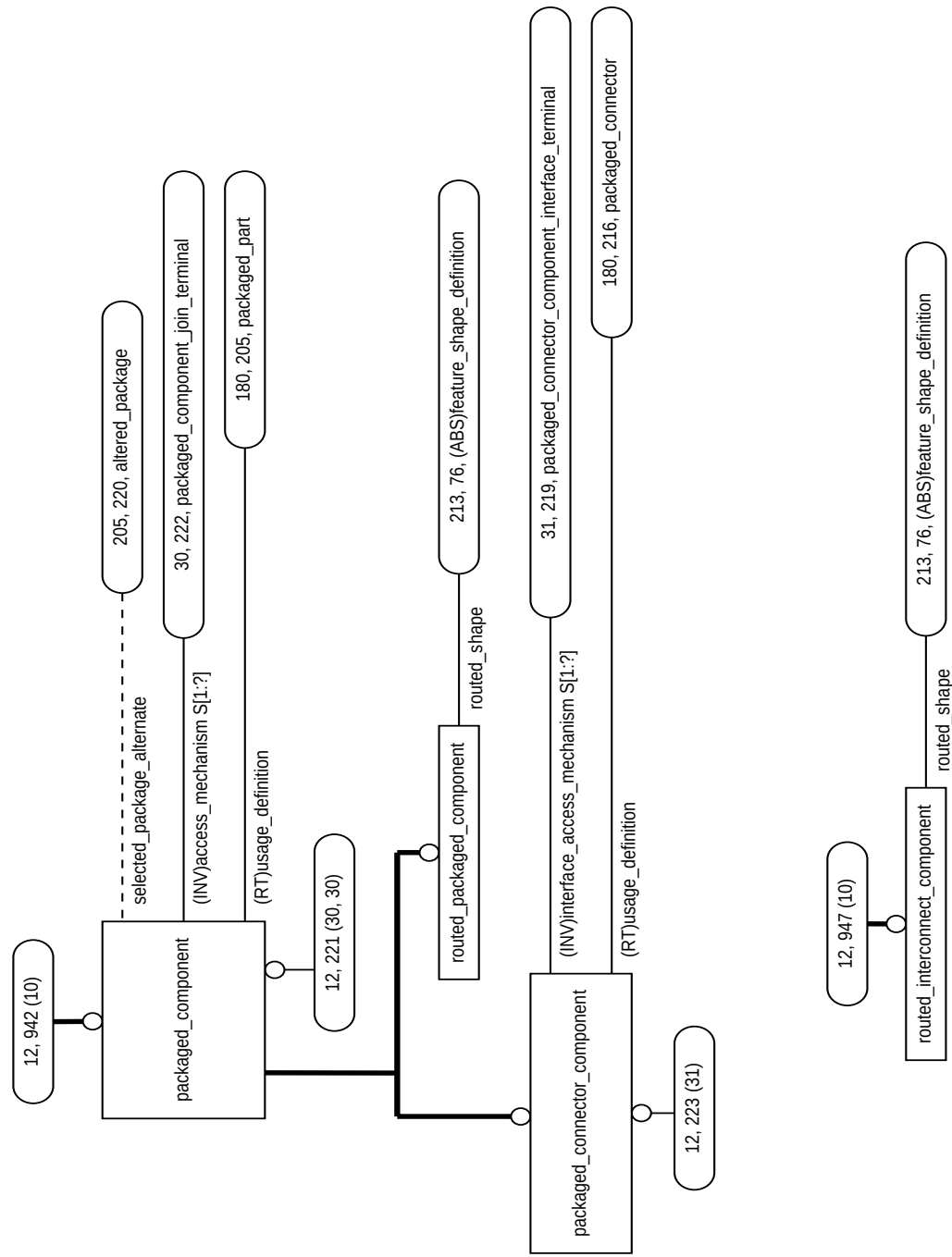


Figure H.12 – ARM EXPRESS-G Graphic 12 of 243

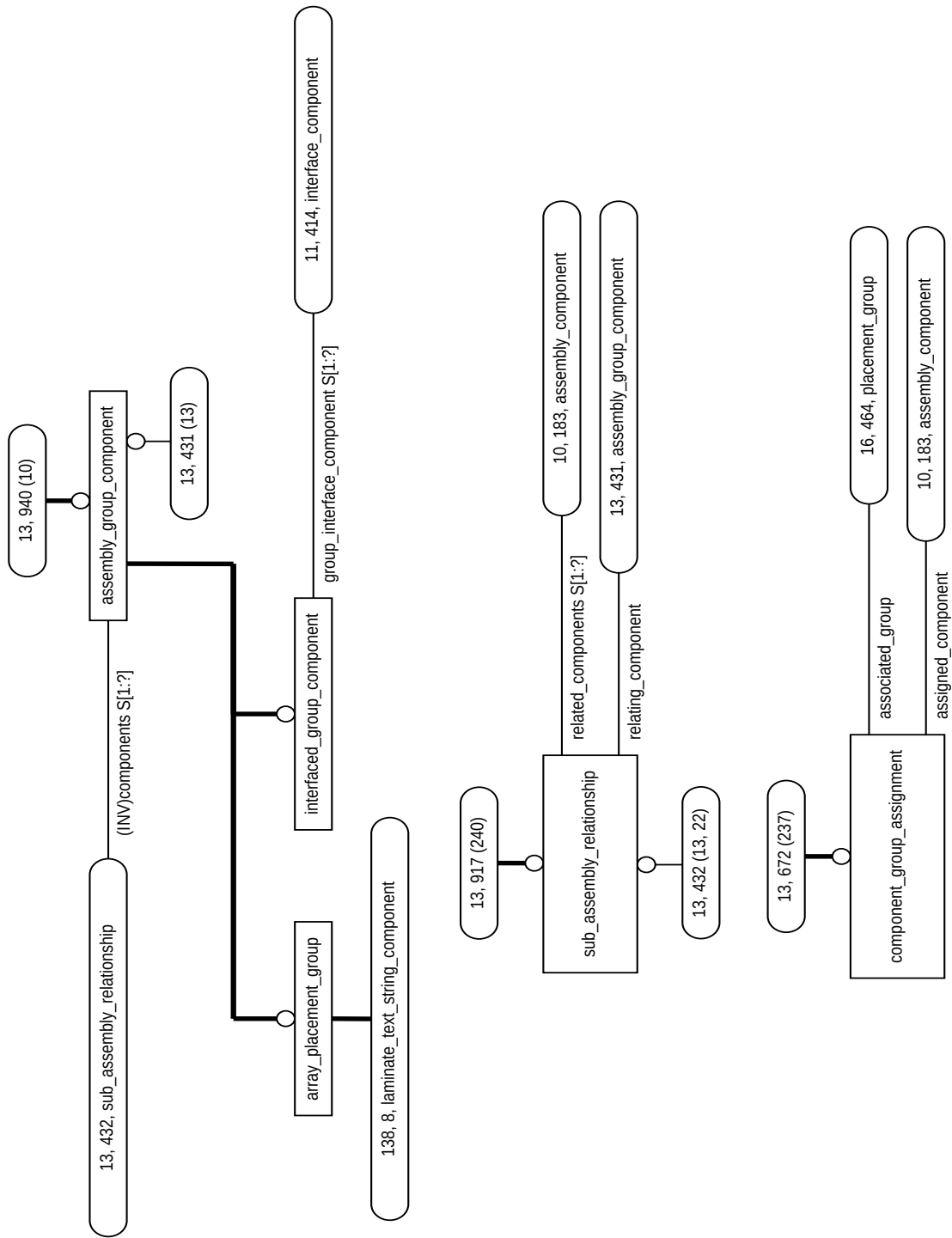


Figure H.13 – ARM EXPRESS-G Graphic 13 of 243

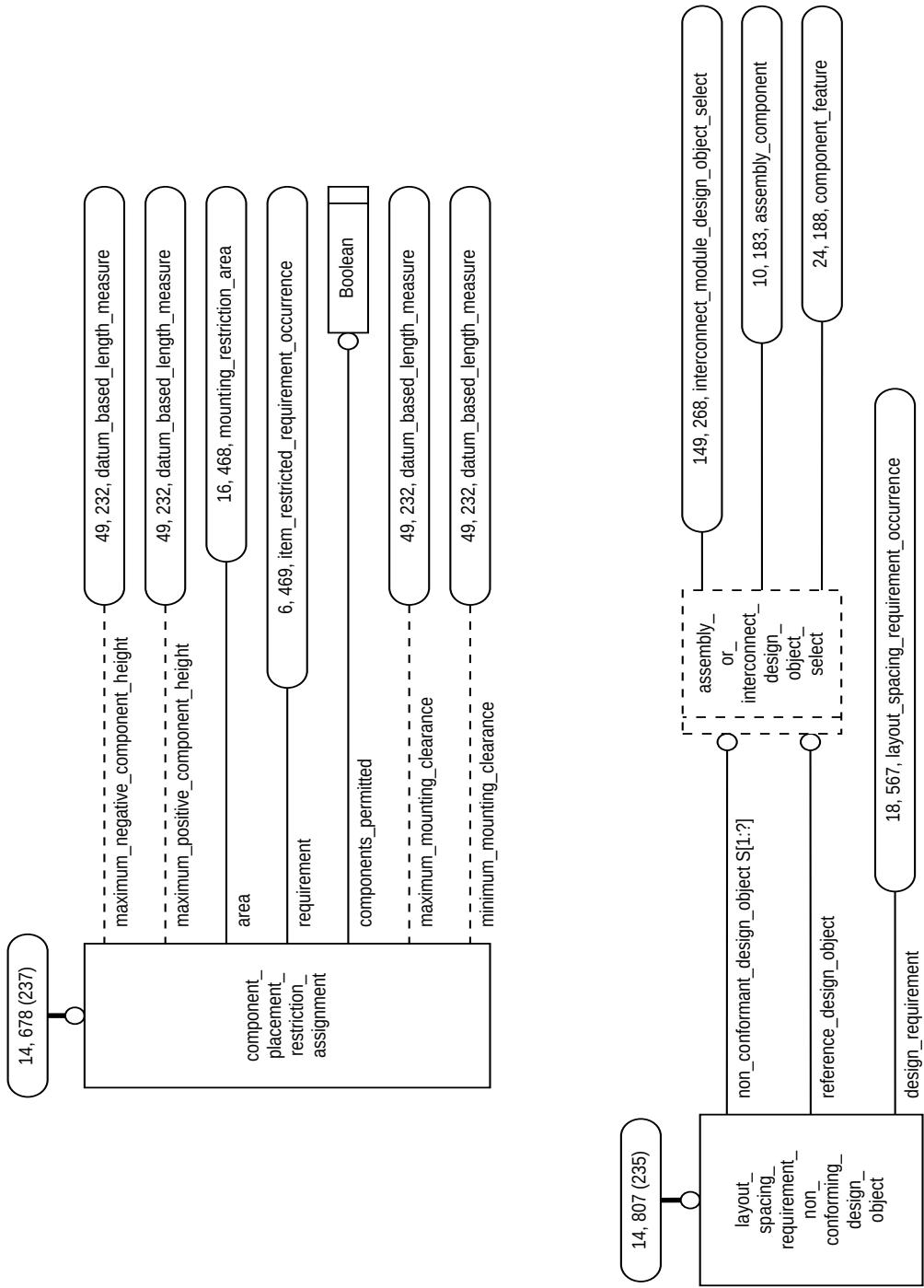


Figure H.14 – ARM EXPRESS-G Graphic 14 of 243

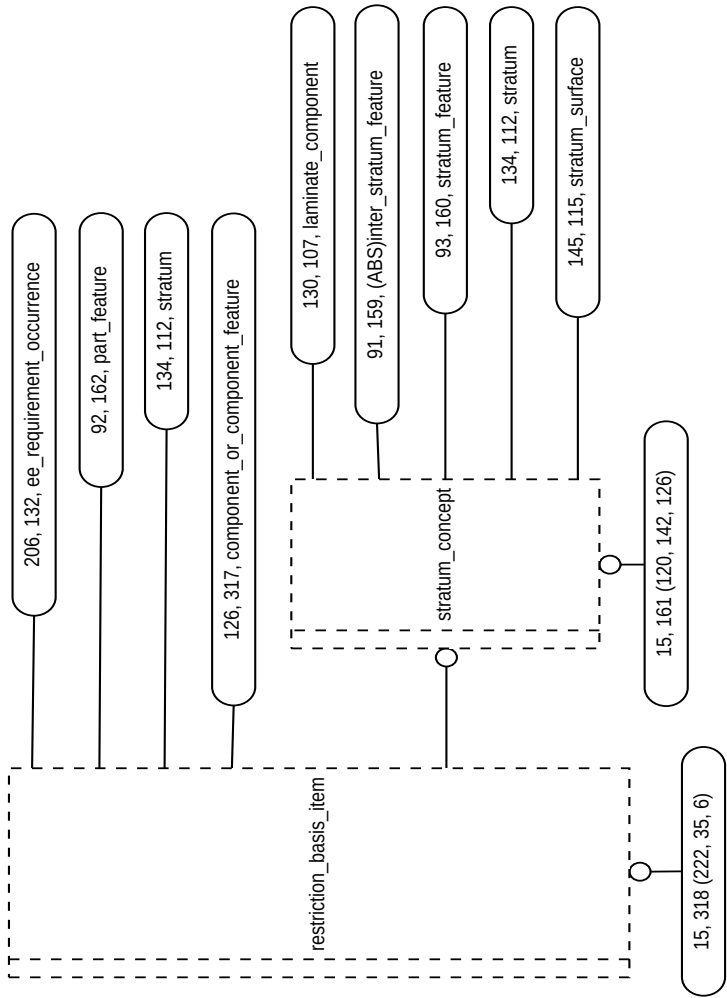


Figure H.15 – ARM EXPRESS-G Graphic 15 of 243

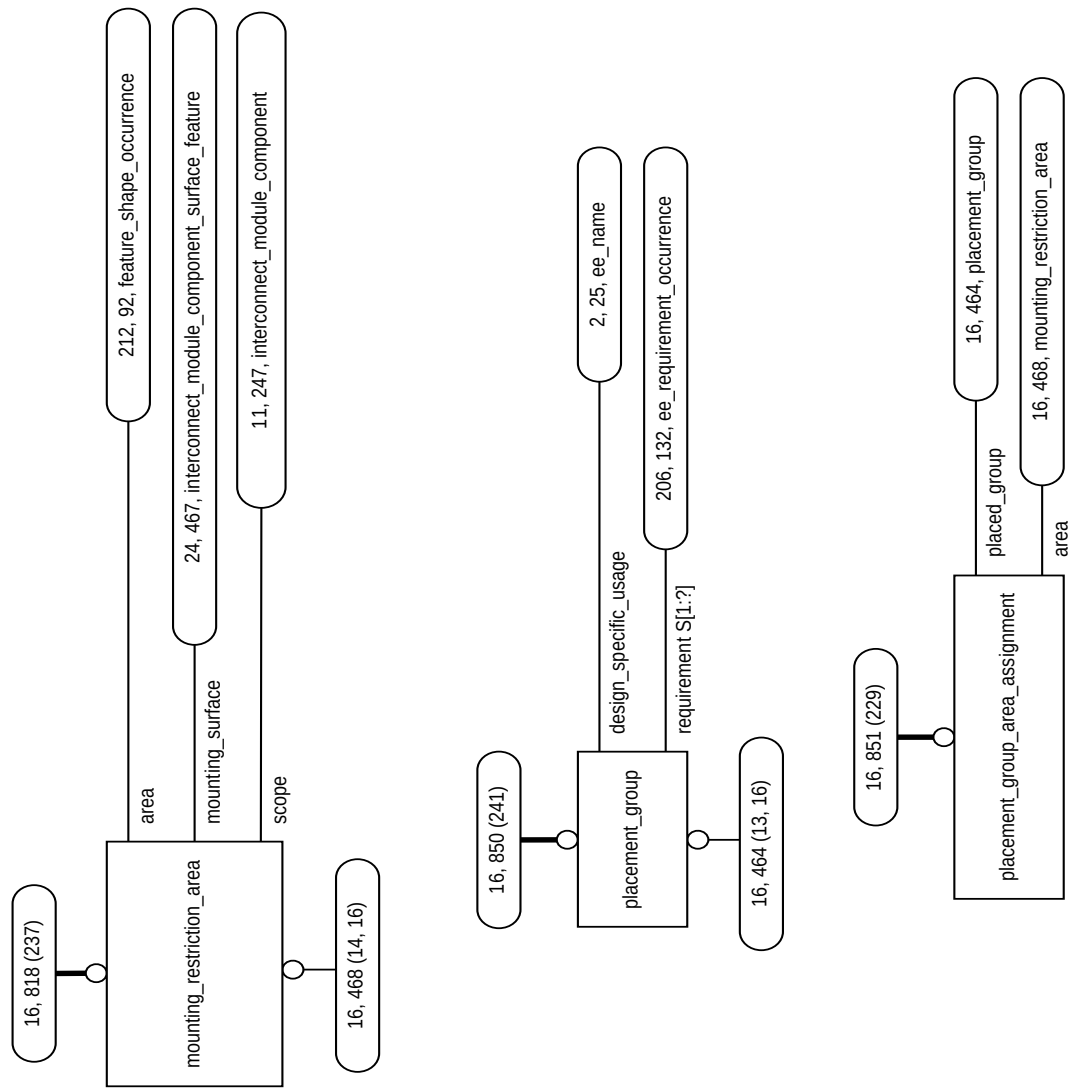


Figure H.16 – ARM EXPRESS-G Graphic 16 of 243

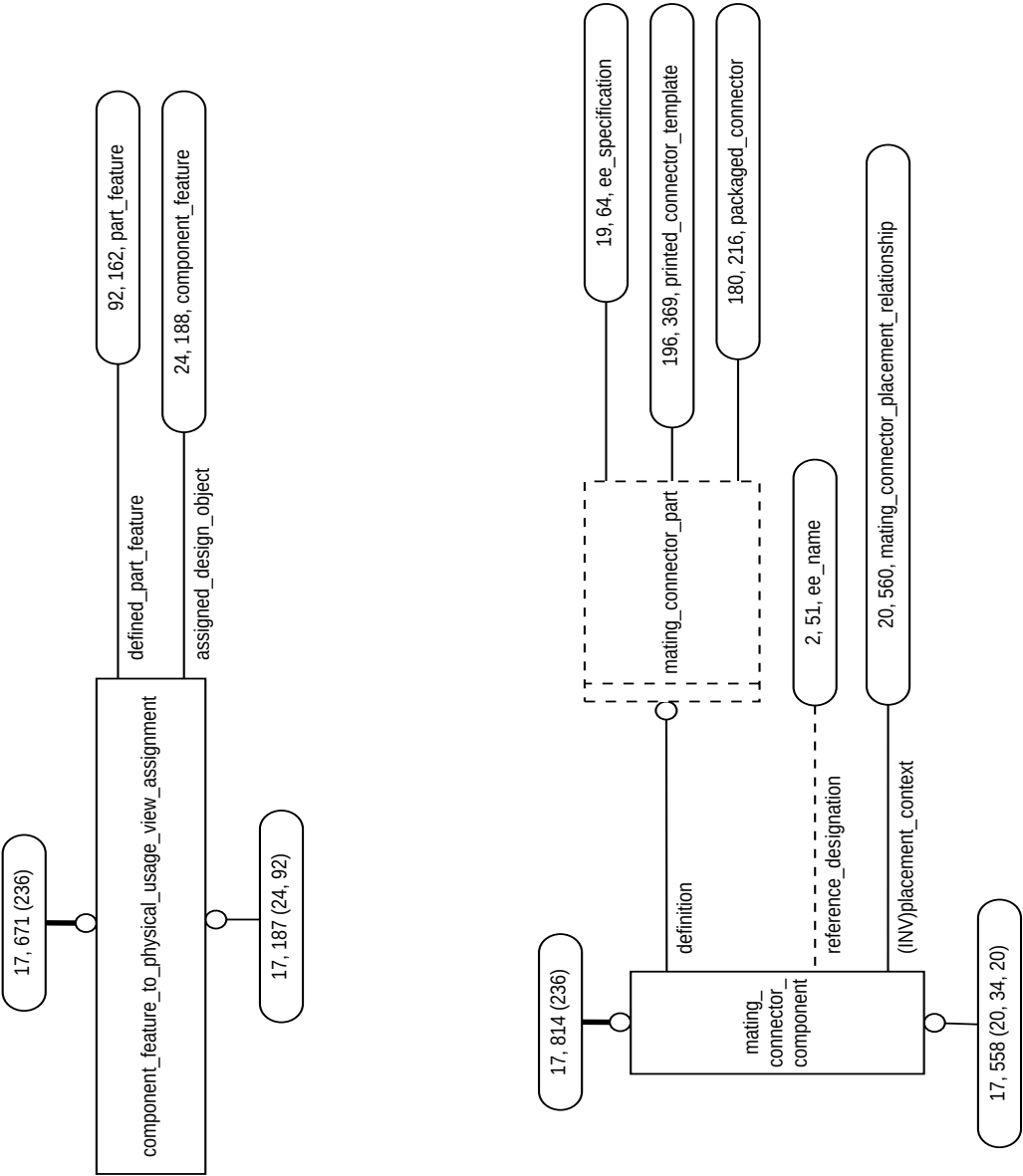


Figure H.17 – ARM EXPRESS-G Graphic 17 of 243

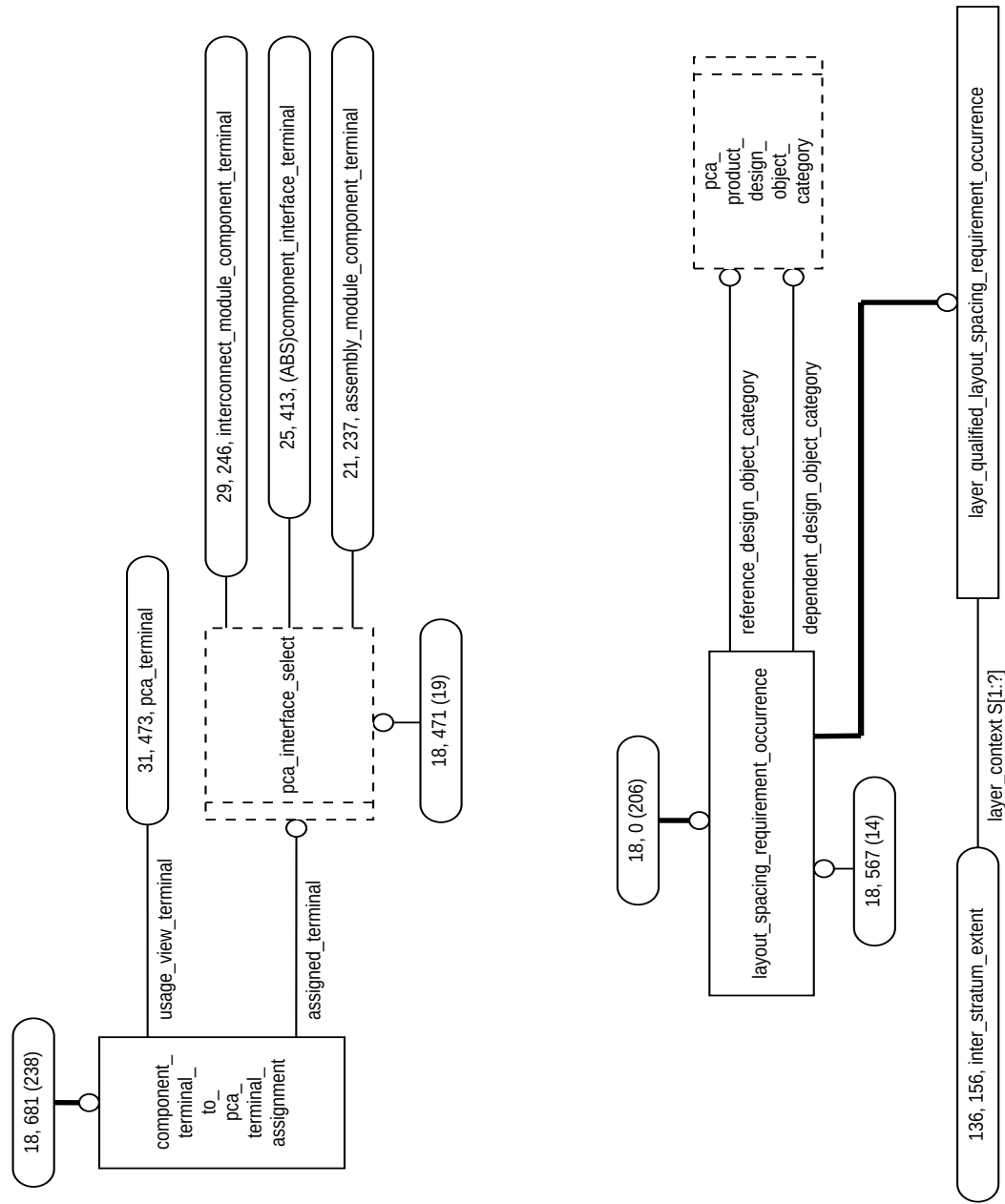


Figure H.18 – ARM EXPRESS-G Graphic 18 of 243

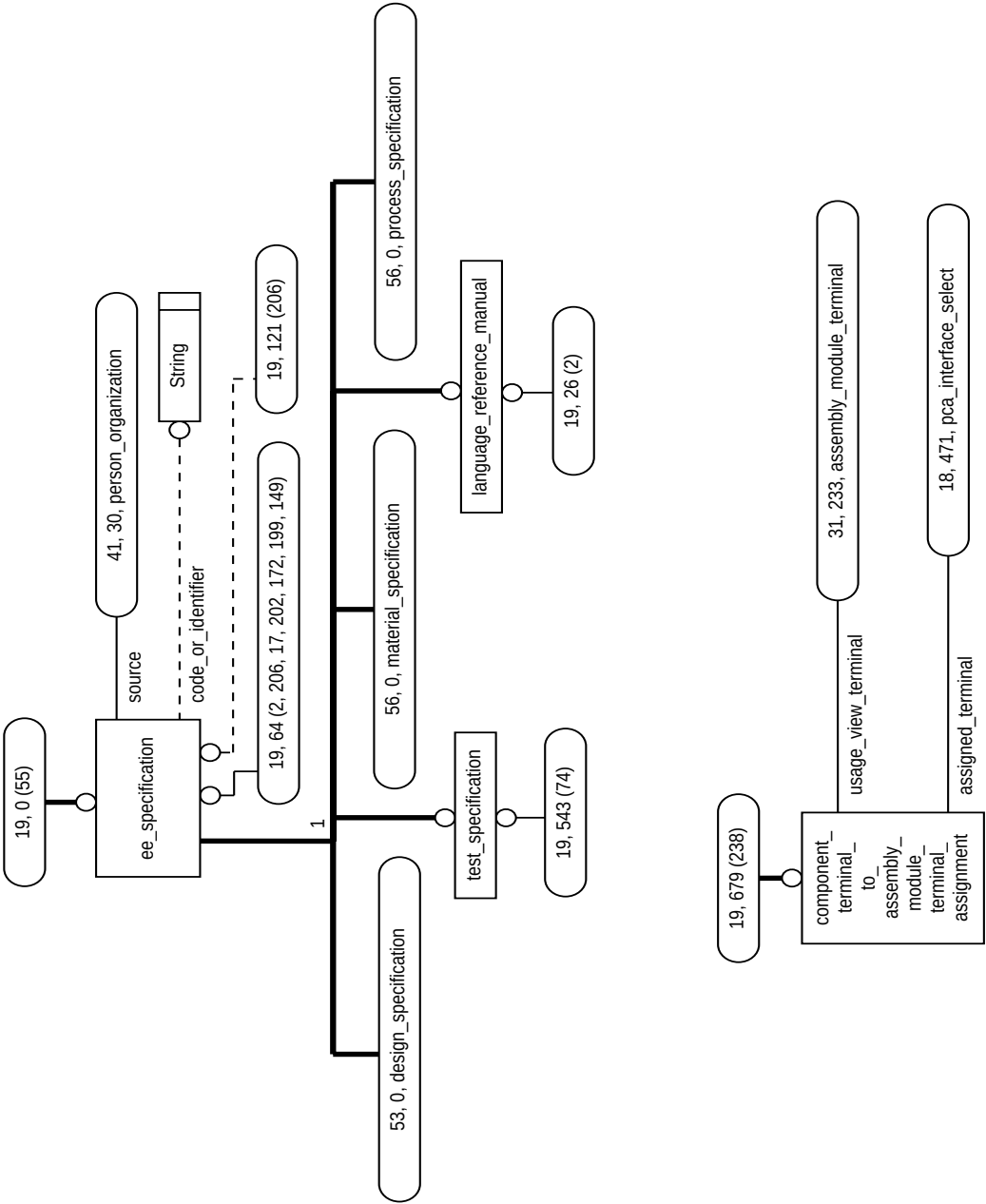


Figure H.19 – ARM EXPRESS-G Graphic 19 of 243

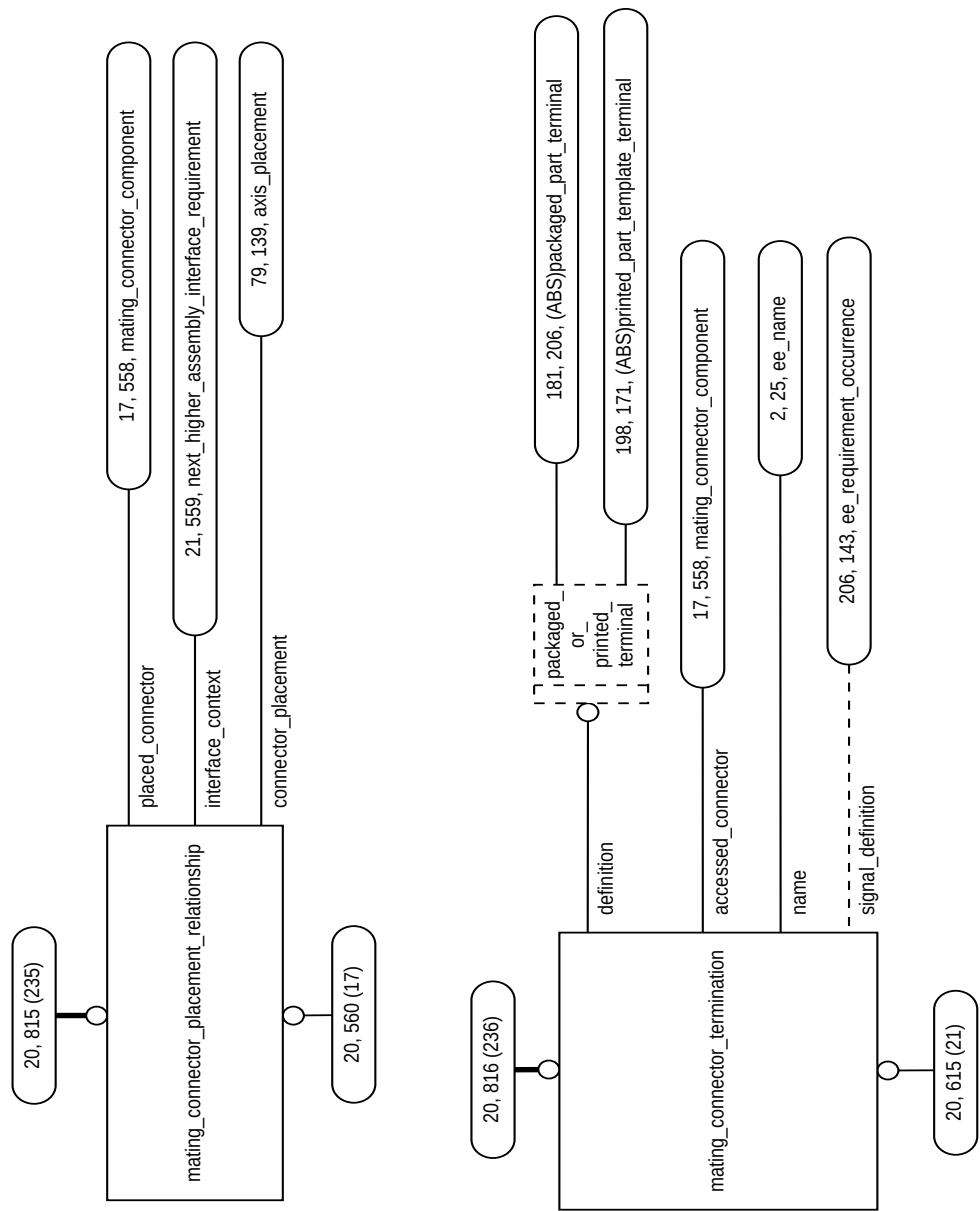


Figure H.20 – ARM EXPRESS-G Graphic 20 of 243

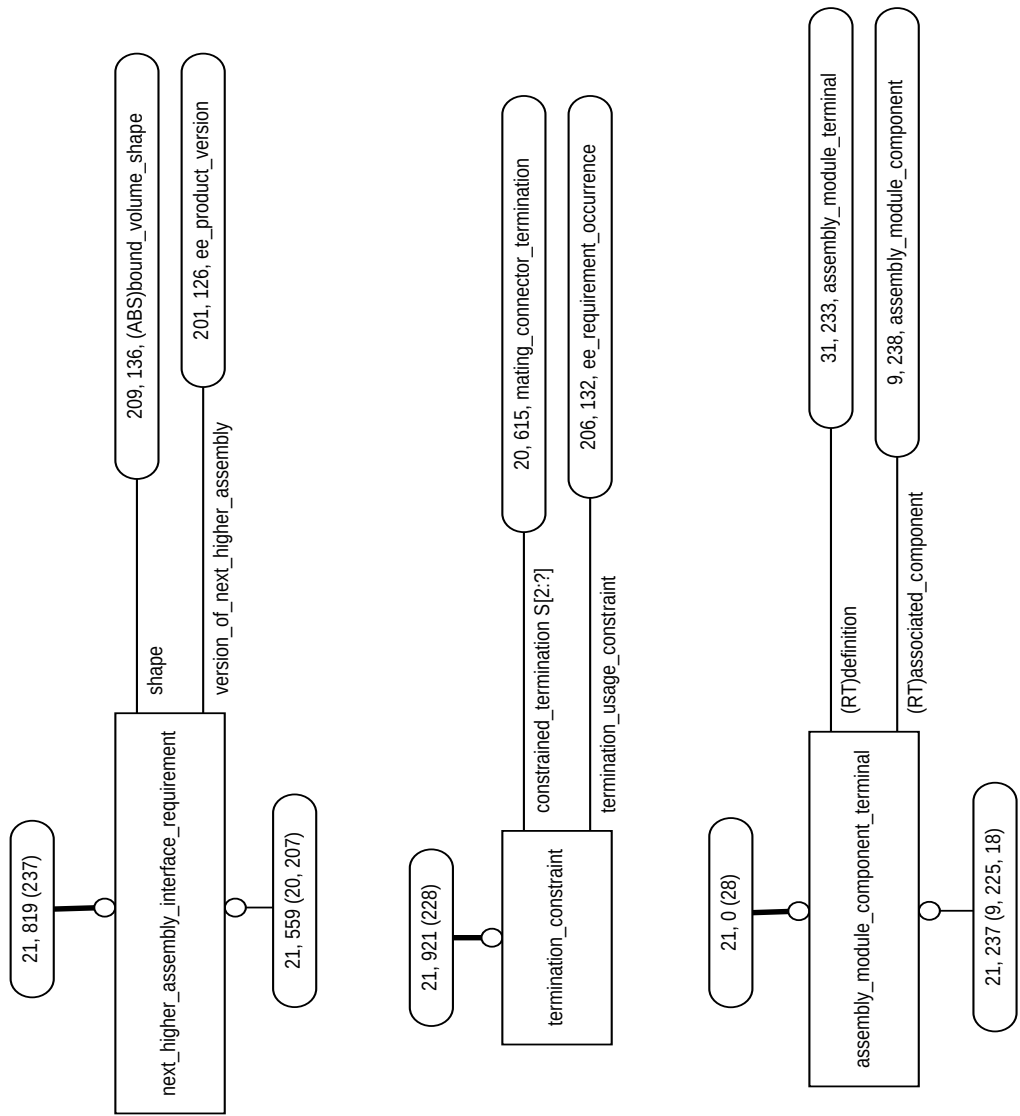


Figure H.21 – ARM EXPRESS-G Graphic 21 of 243

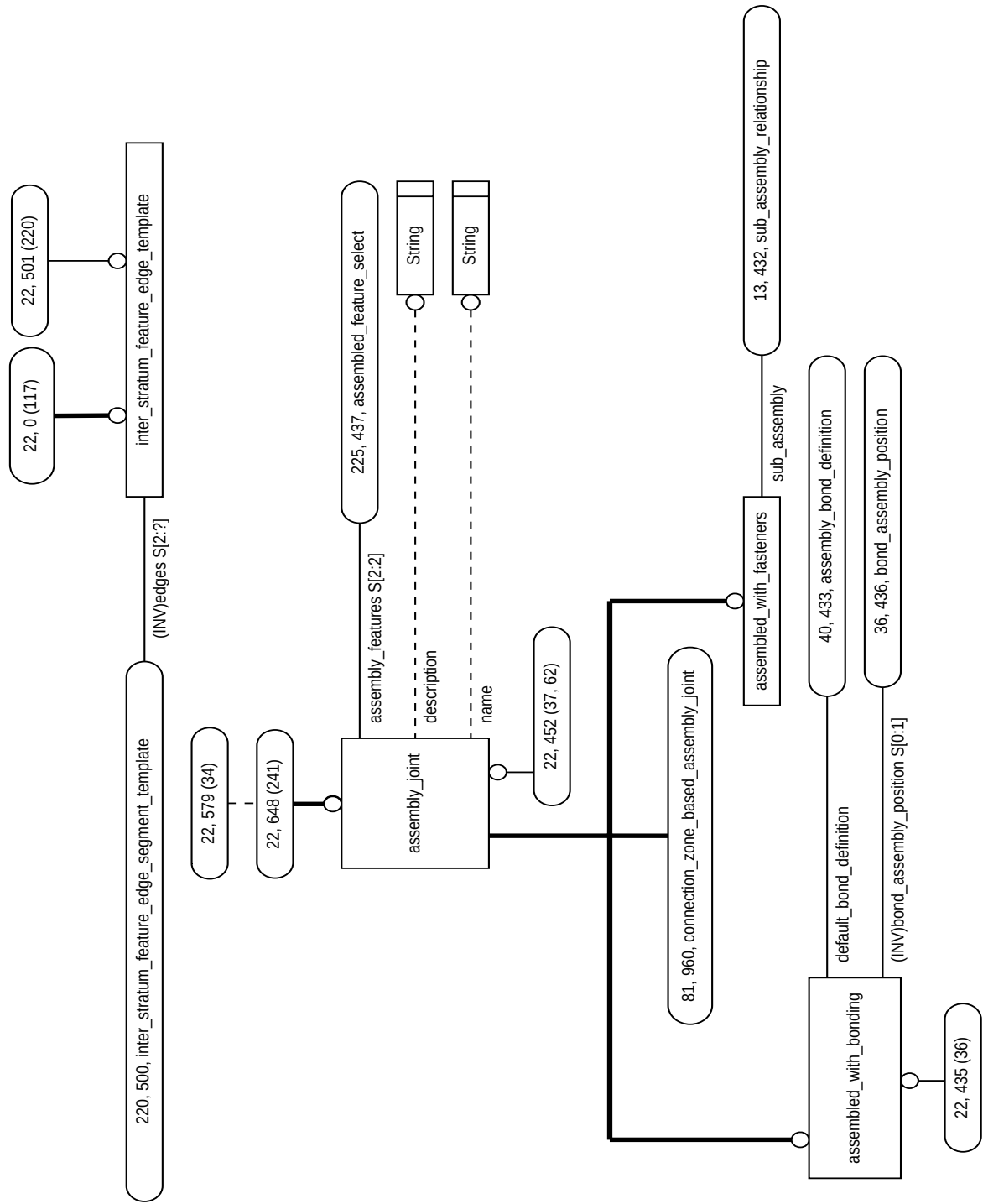


Figure H.22 – ARM EXPRESS-G Graphic 22 of 243

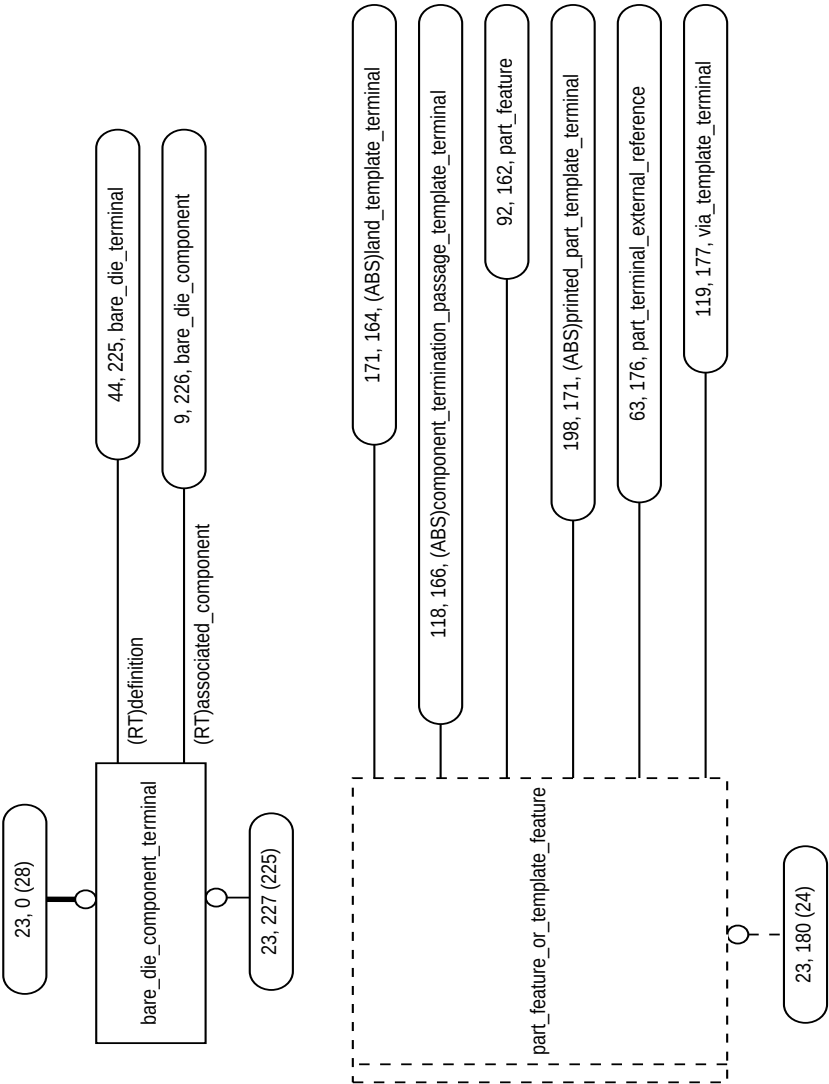


Figure H.23 – ARM EXPRESS-G Graphic 23 of 243

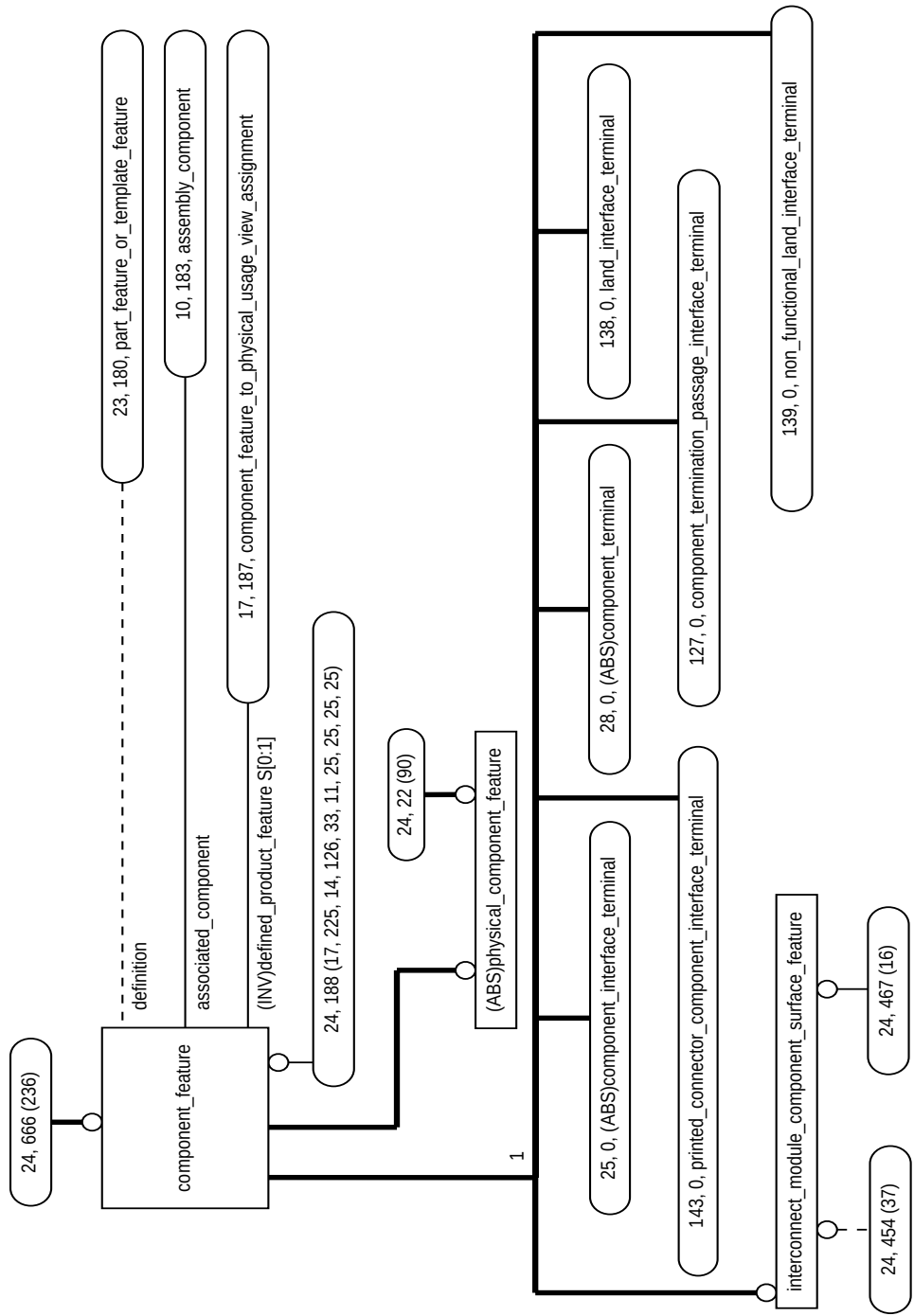


Figure H.24 – ARM EXPRESS-G Graphic 24 of 243

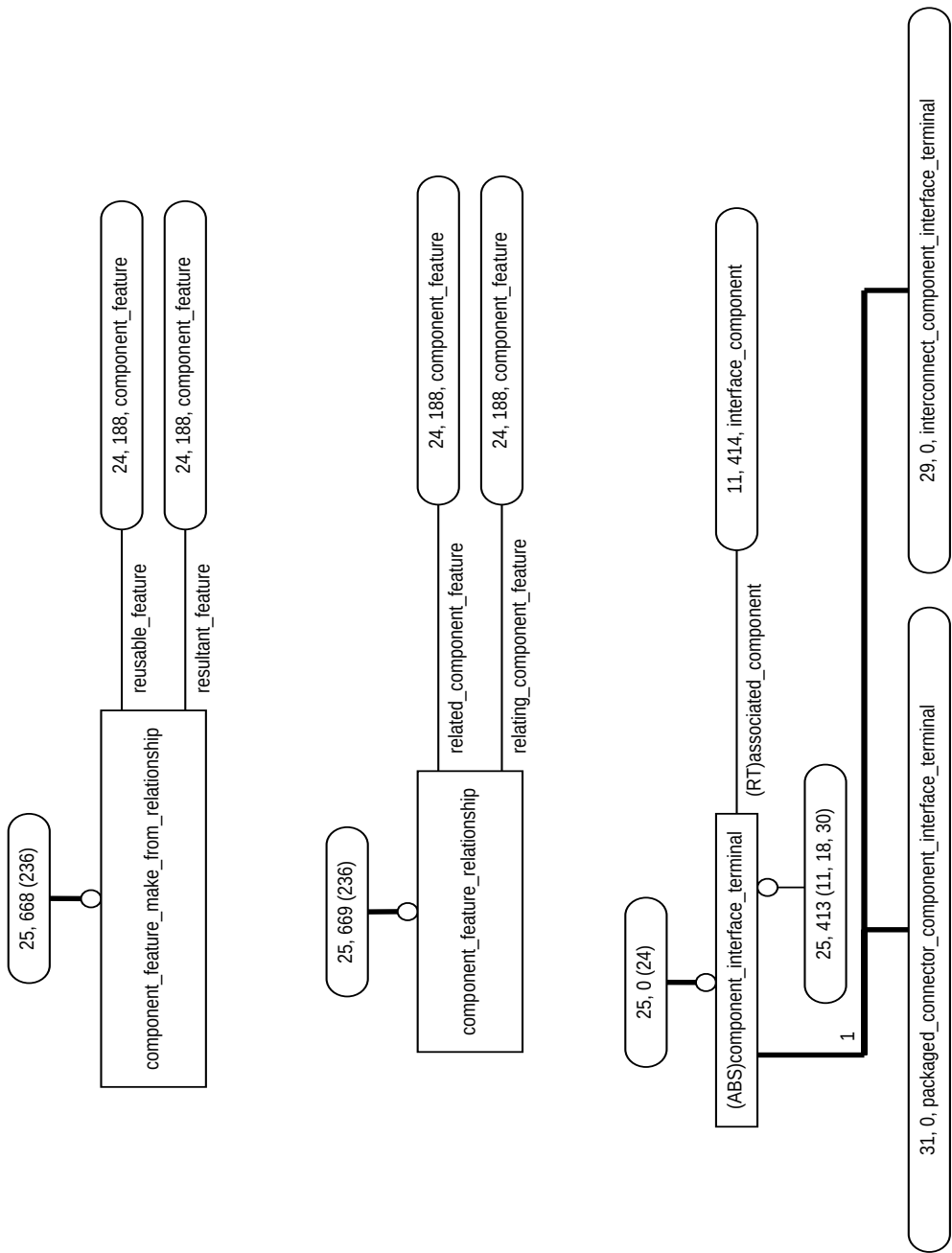


Figure H.25 – ARM EXPRESS-G Graphic 25 of 243

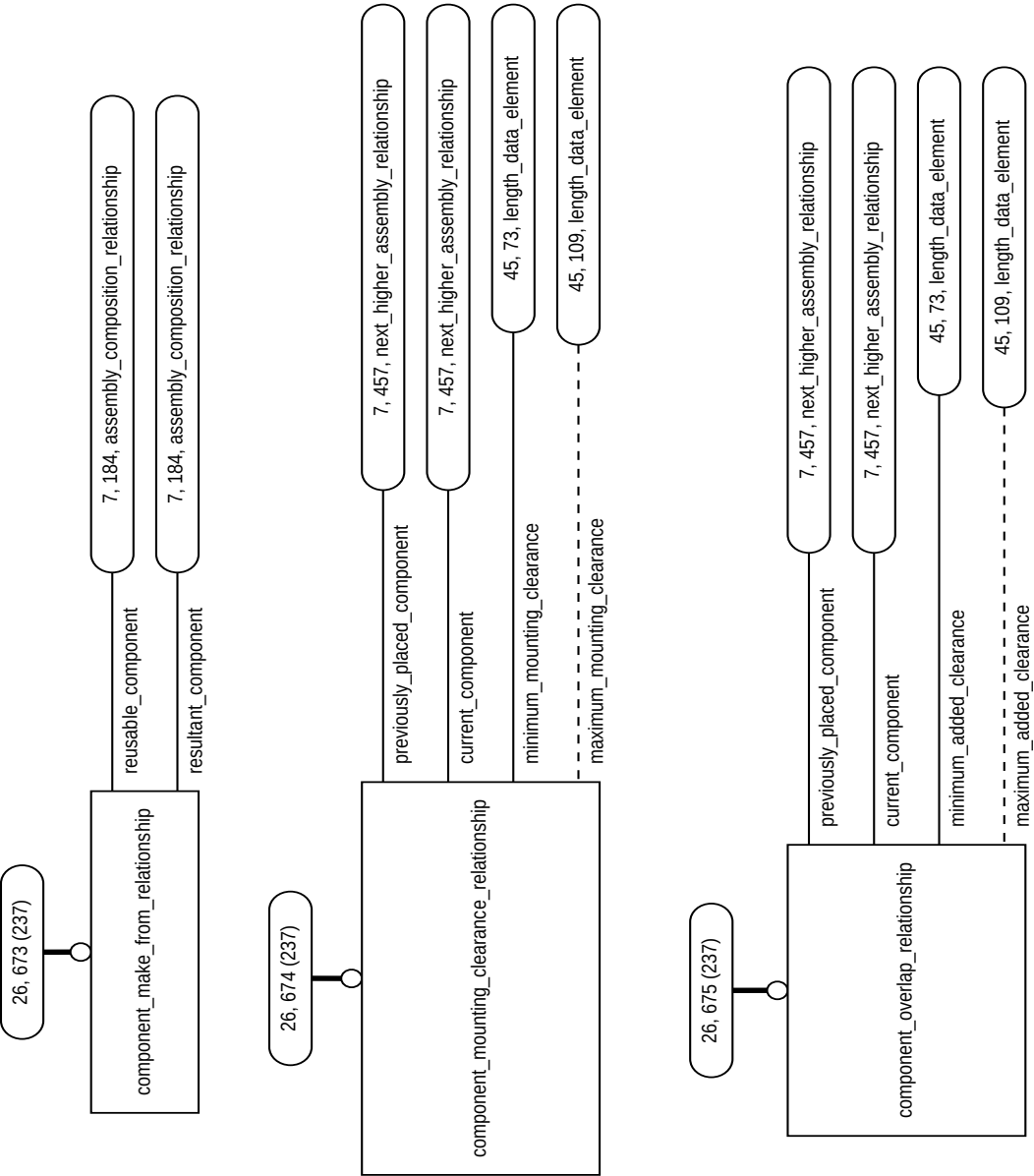


Figure H.26 – ARM EXPRESS-G Graphic 26 of 243

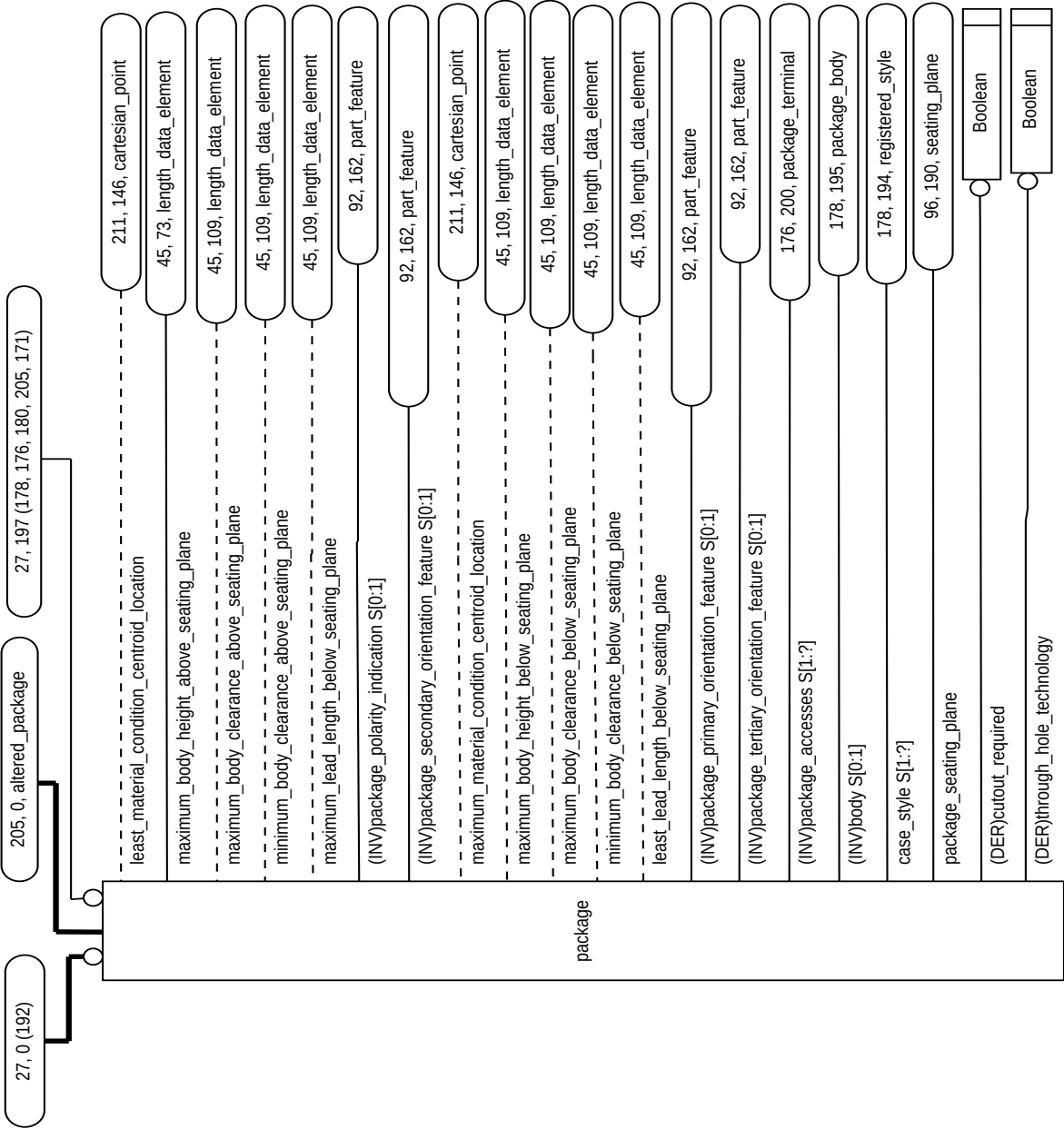


Figure H.27 – ARM EXPRESS-G Graphic 27 of 243

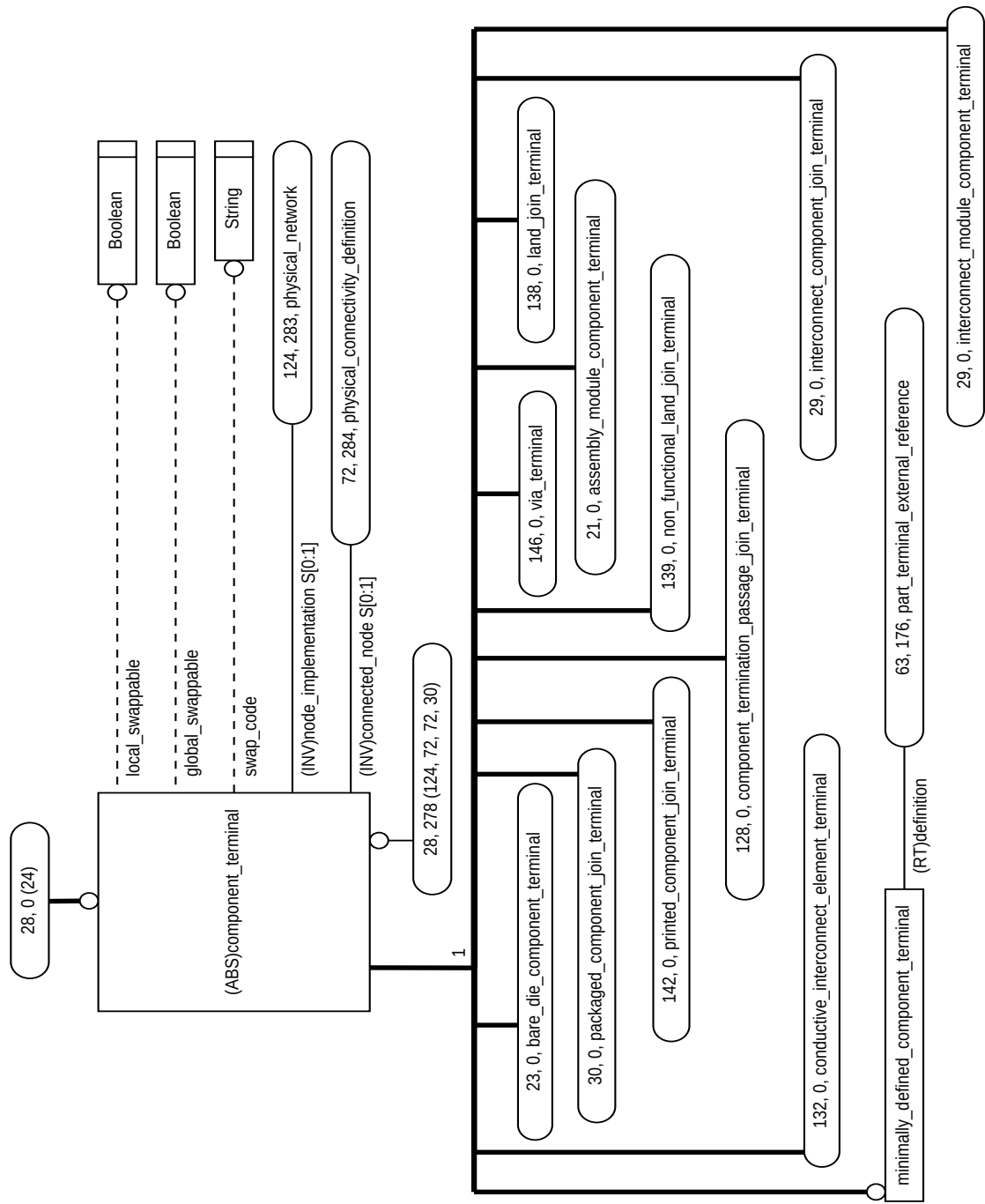


Figure H.28 – ARM EXPRESS-G Graphic 28 of 243

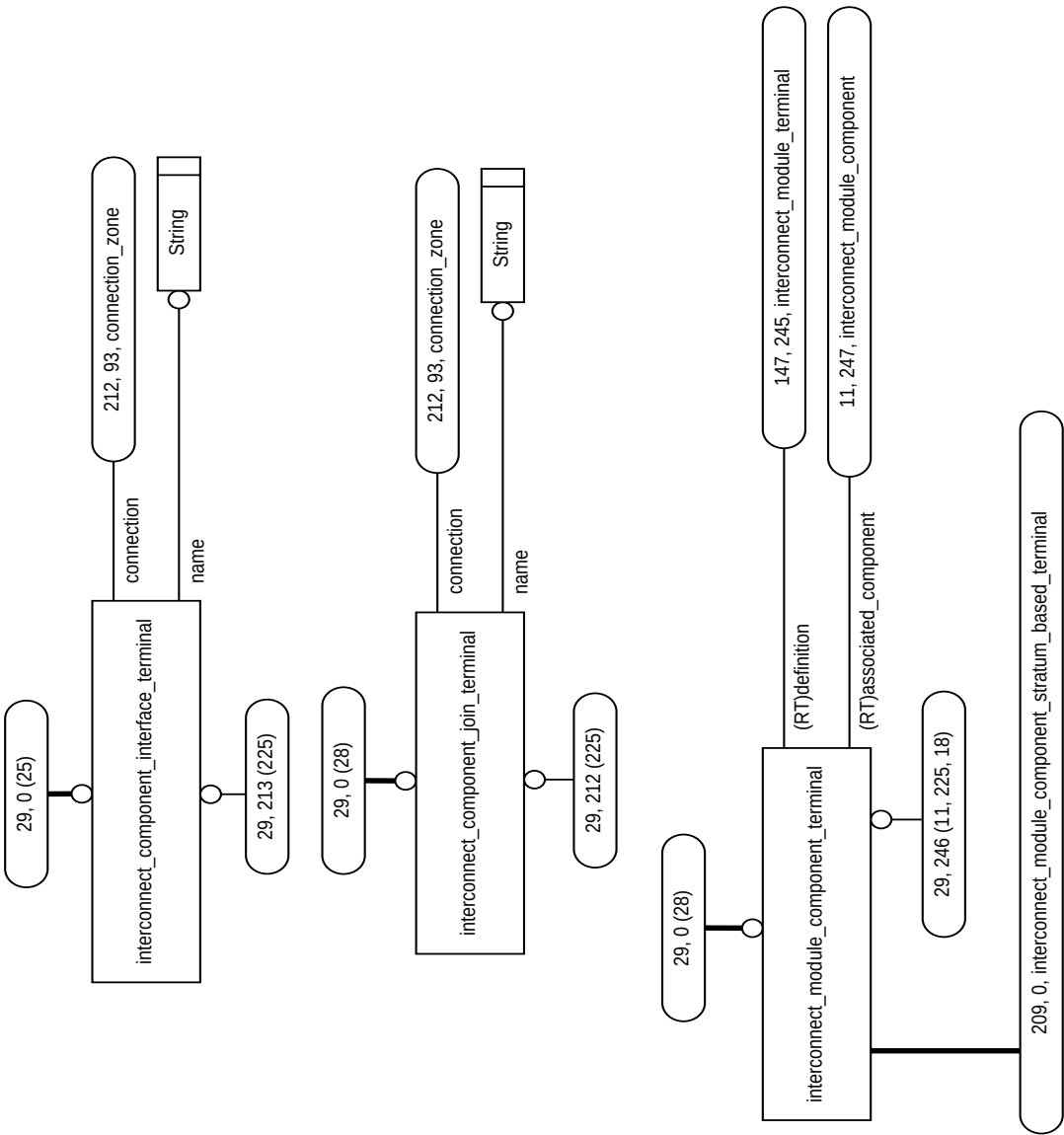


Figure H.29 – ARM EXPRESS-G Graphic 29 of 243

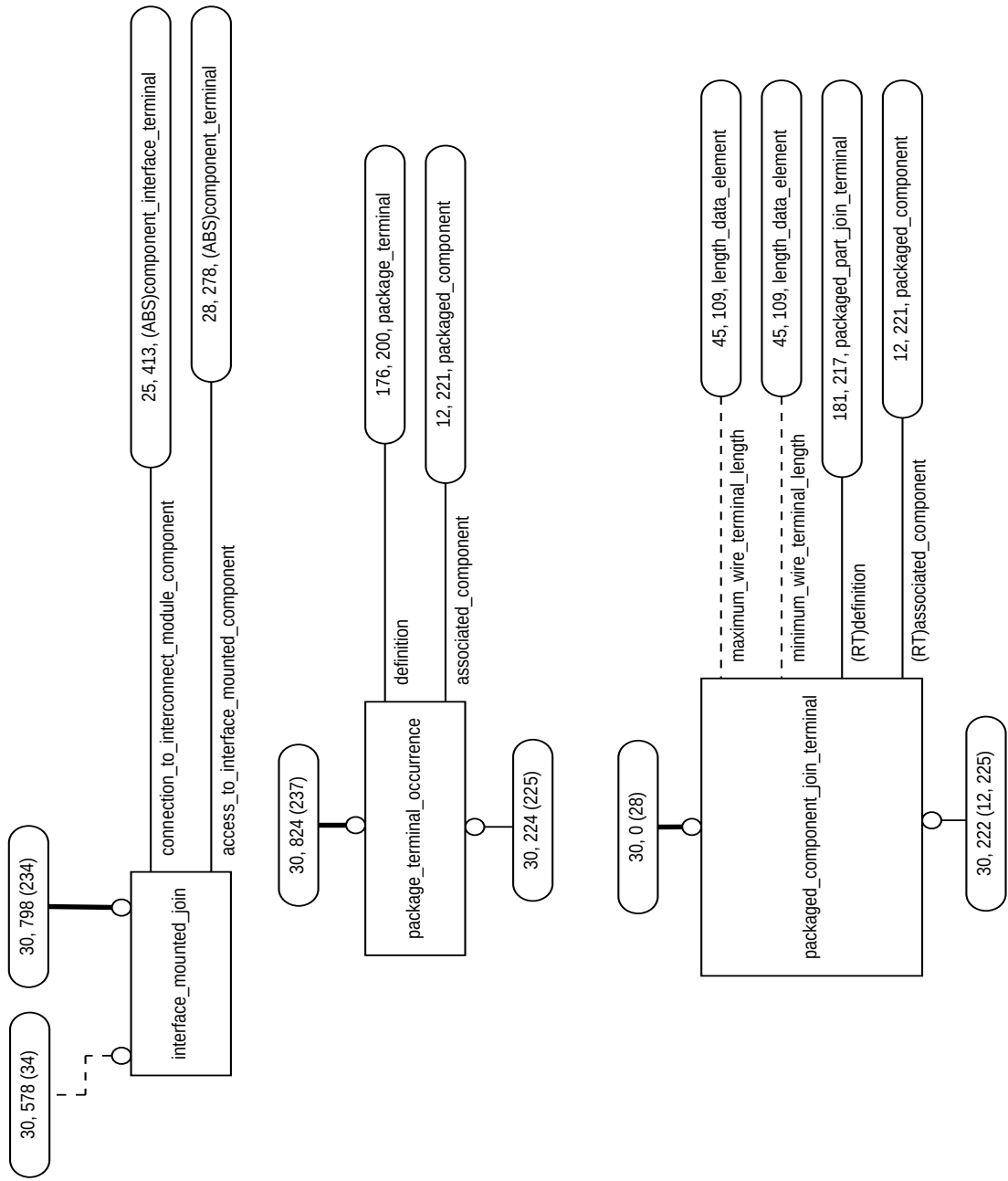


Figure H.30 – ARM EXPRESS-G Graphic 30 of 243

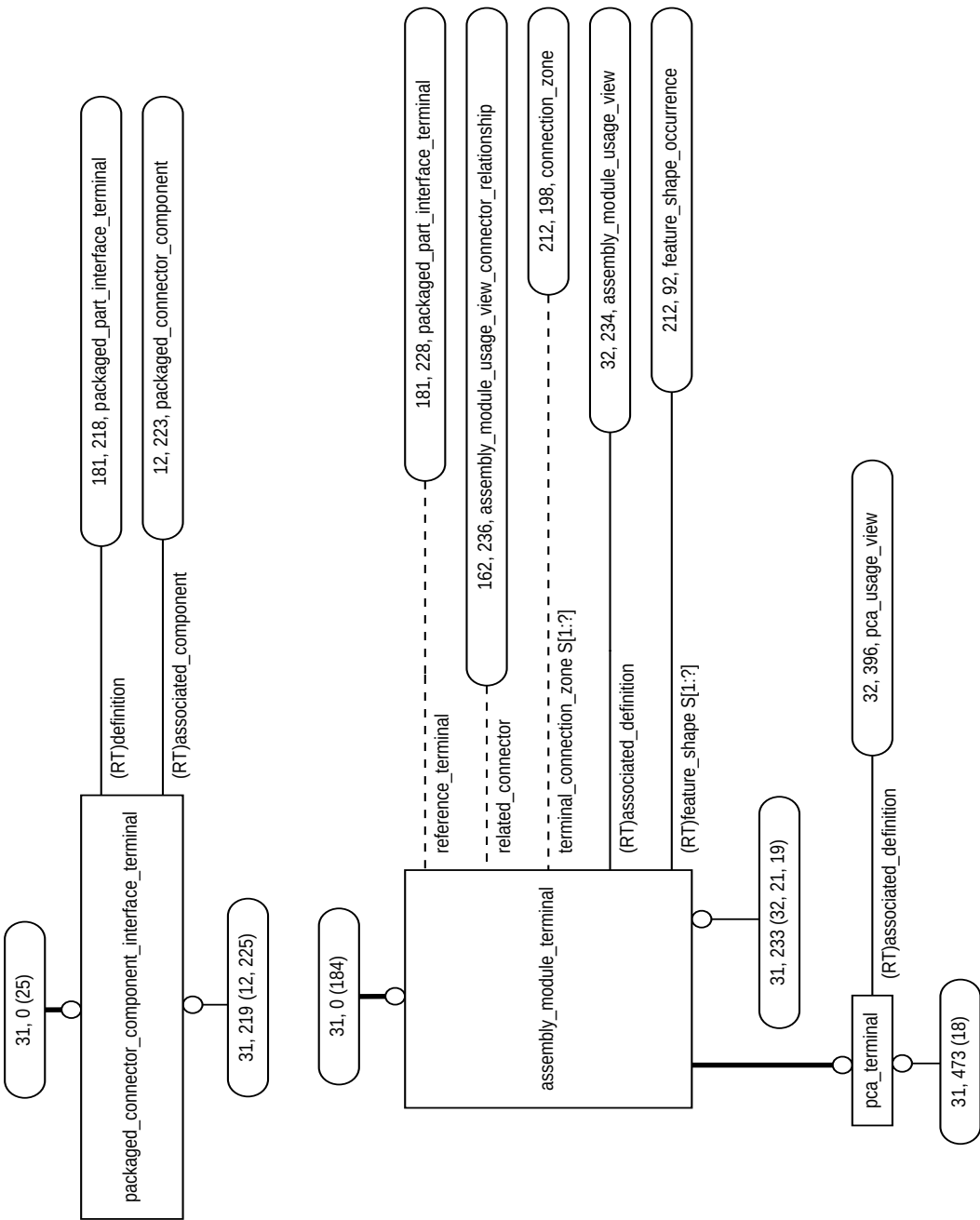


Figure H.31 – ARM EXPRESS-G Graphic 31 of 243

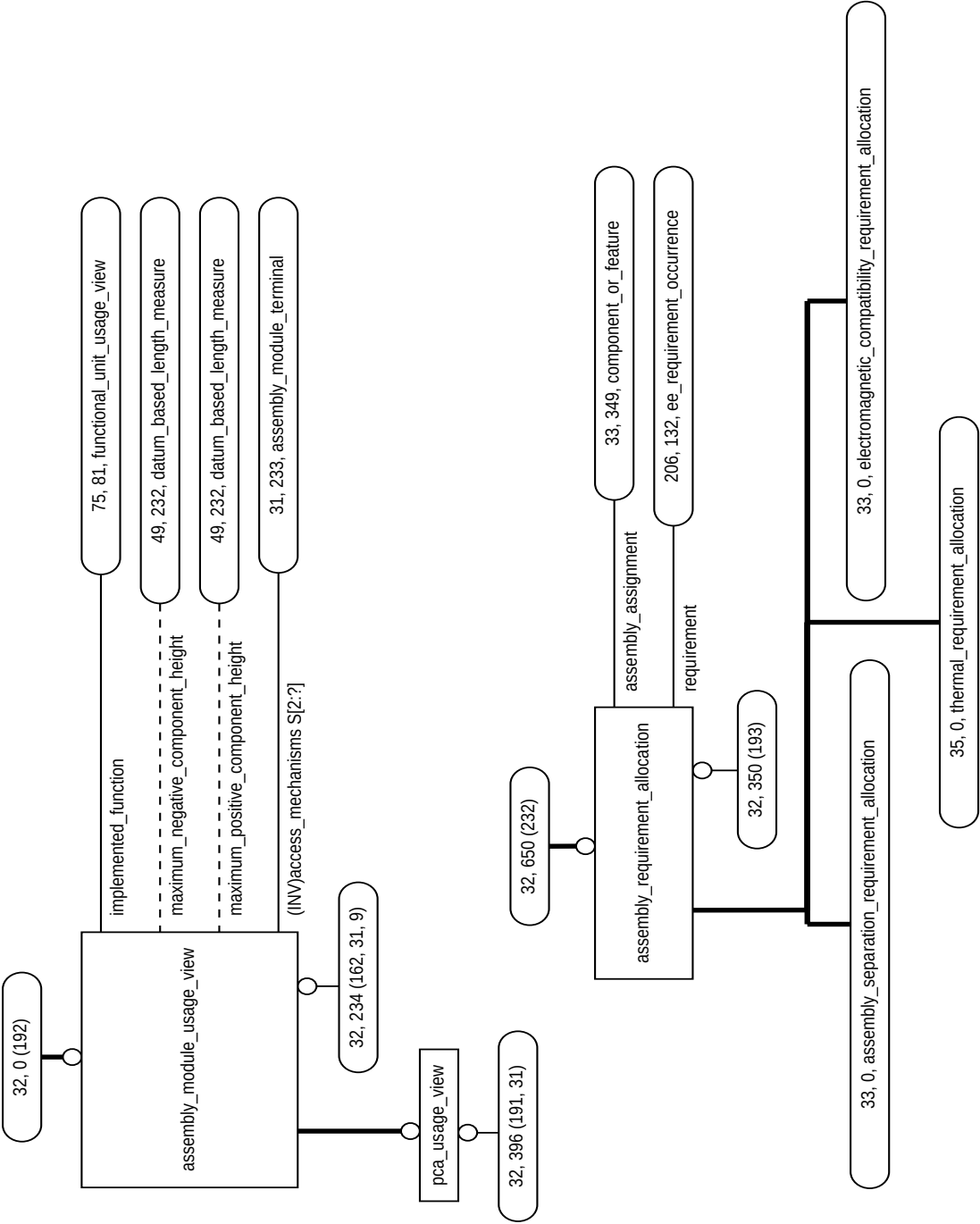


Figure H.32 – ARM EXPRESS-G Graphic 32 of 243

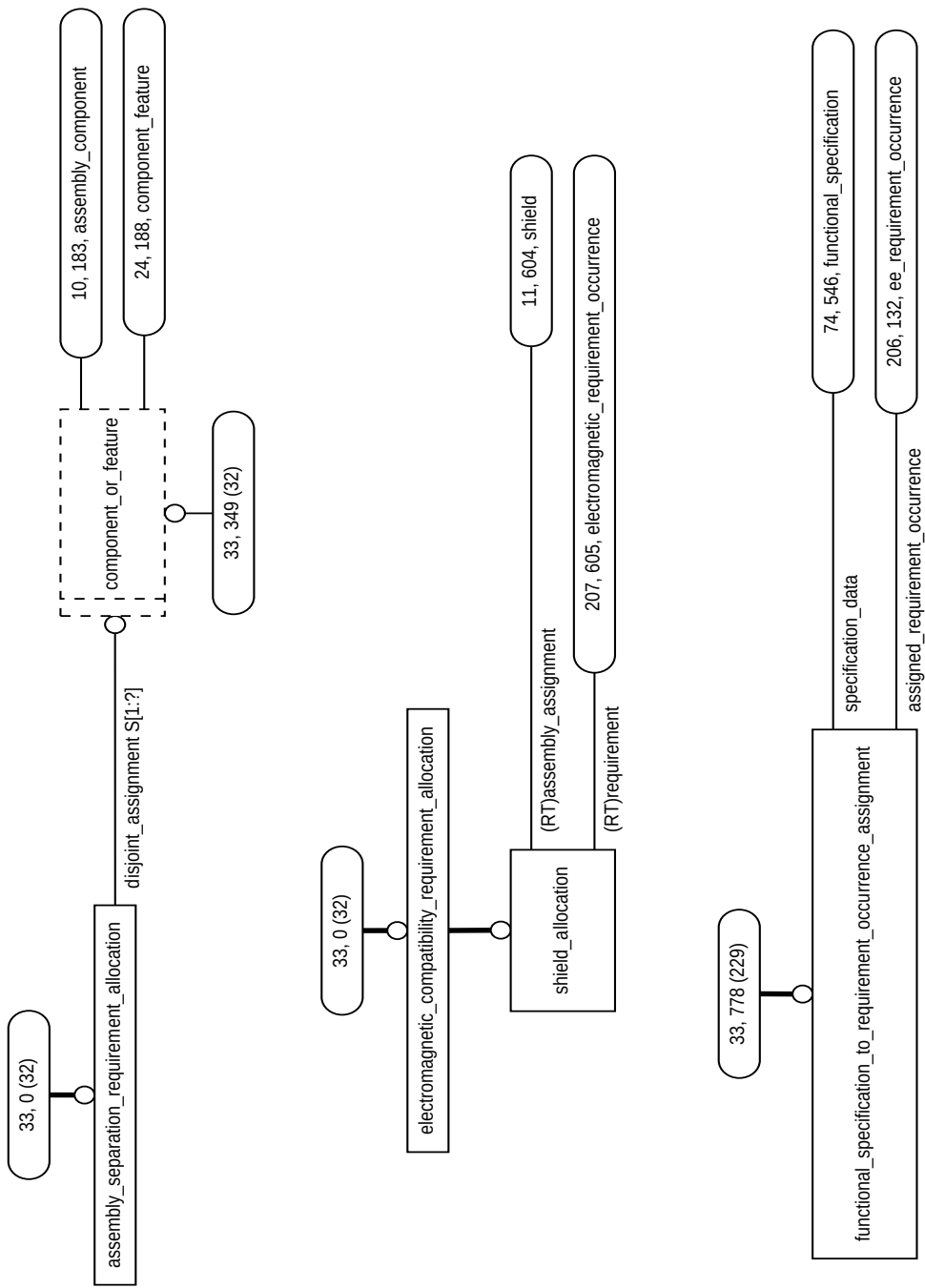


Figure H.33 – ARM EXPRESS-G Graphic 33 of 243



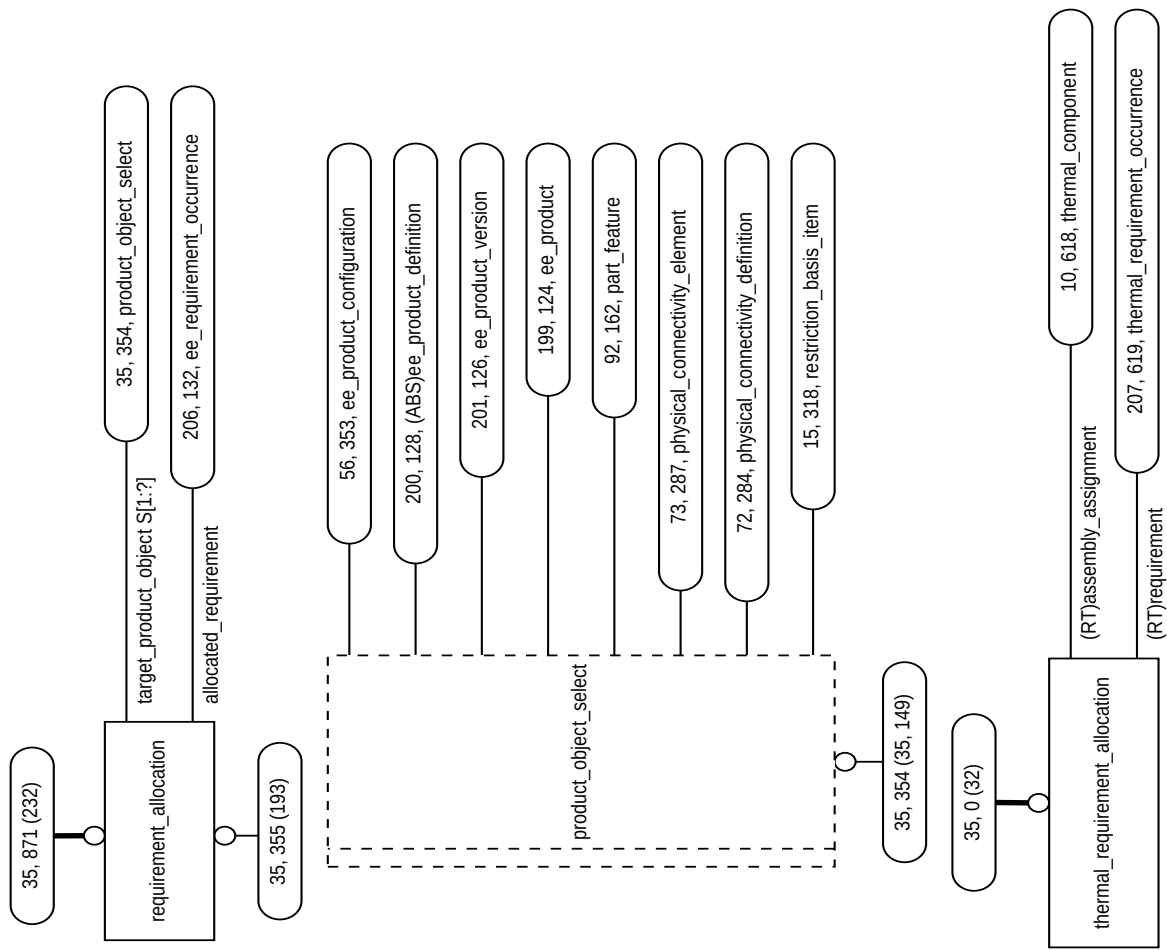


Figure H.35 – ARM EXPRESS-G Graphic 35 of 243

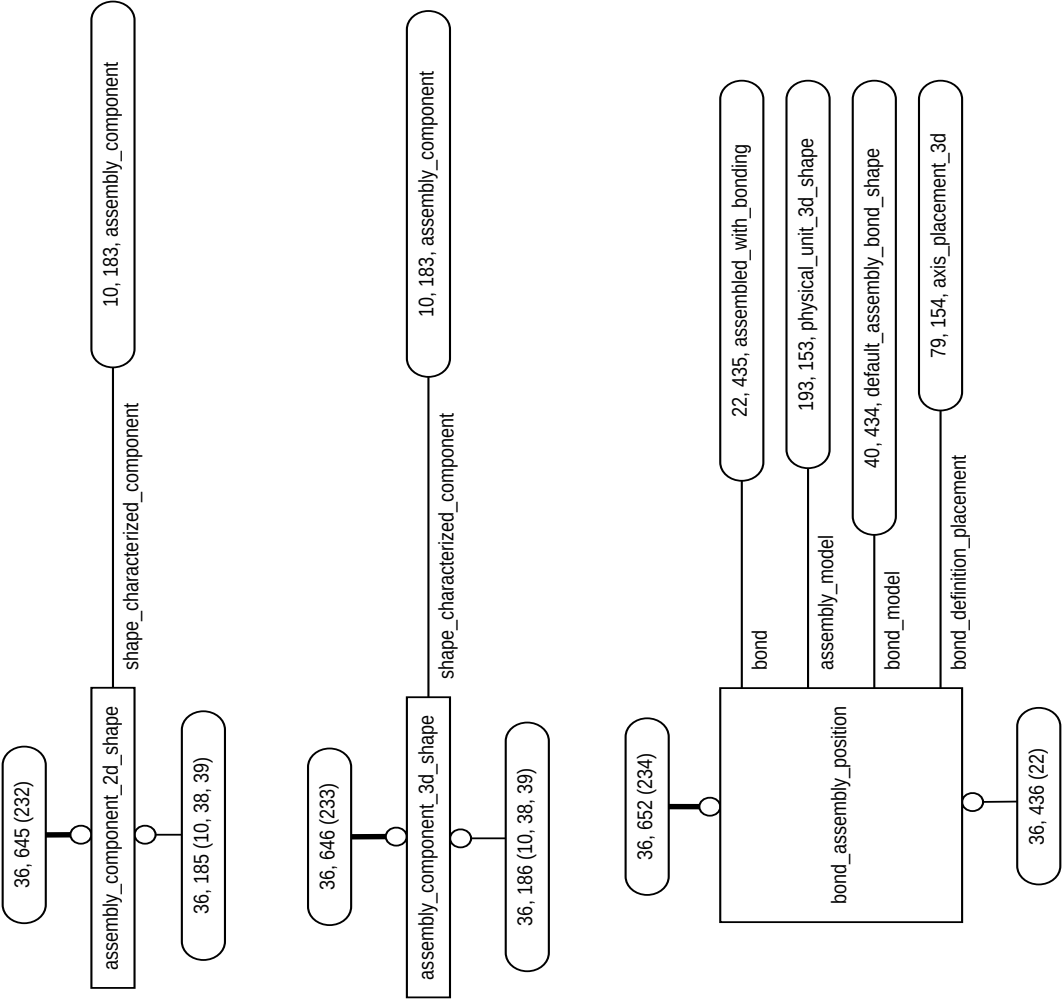


Figure H.36 – ARM EXPRESS-G Graphic 36 of 243

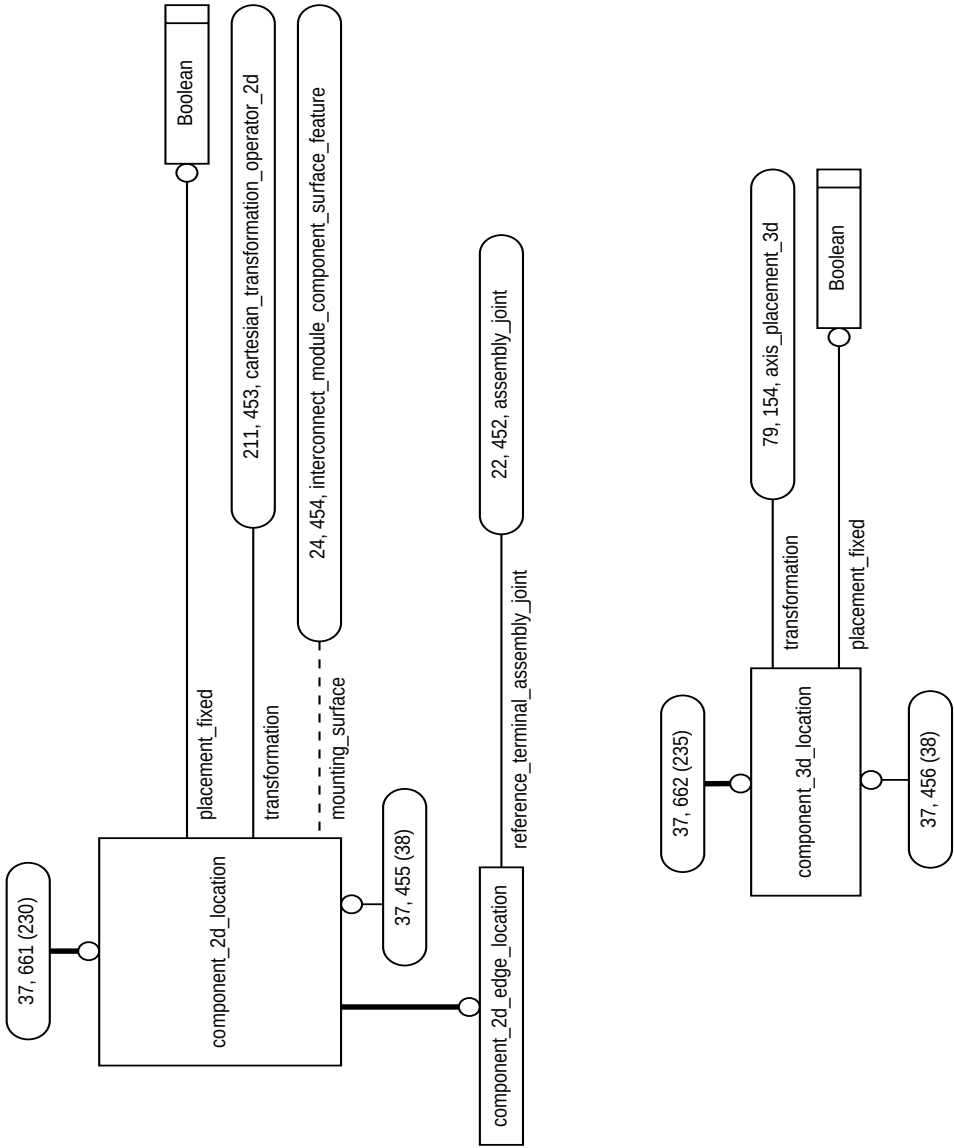


Figure H.37 – ARM EXPRESS-G Graphic 37 of 243

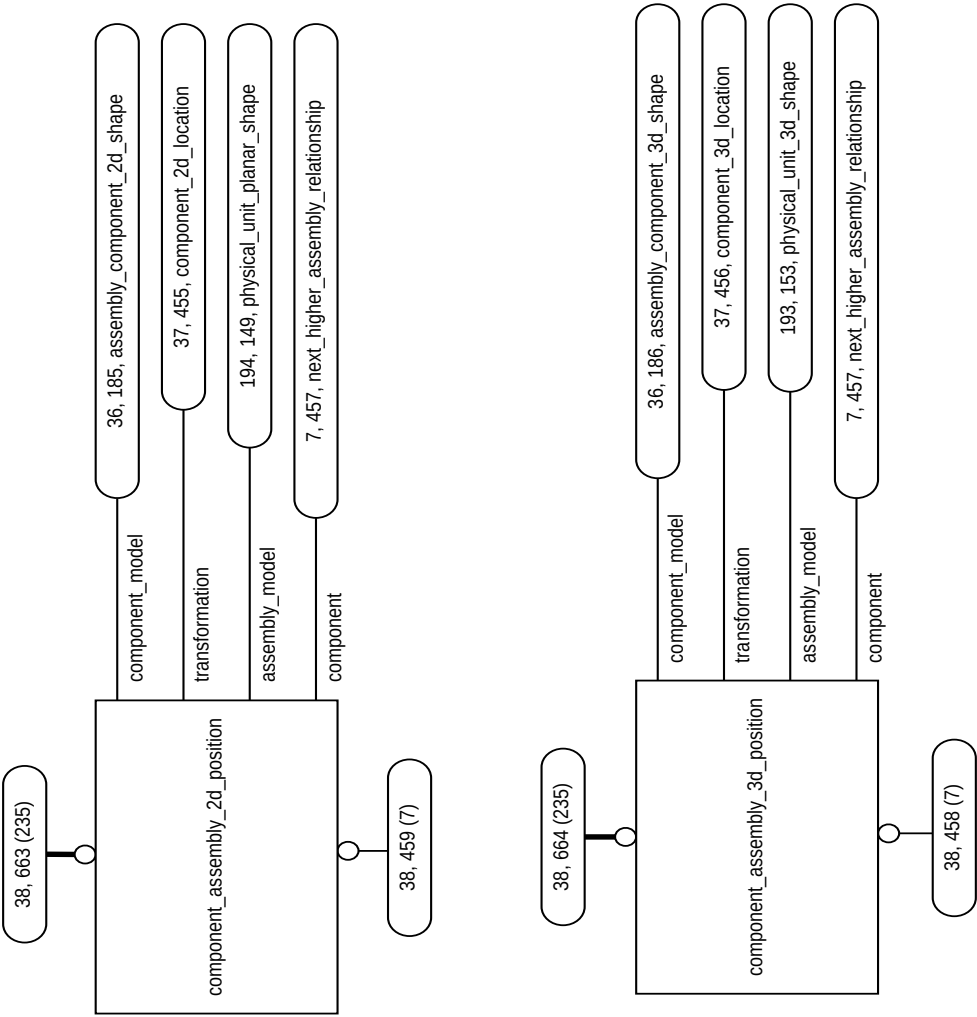


Figure H.38 – ARM EXPRESS-G Graphic 38 of 243

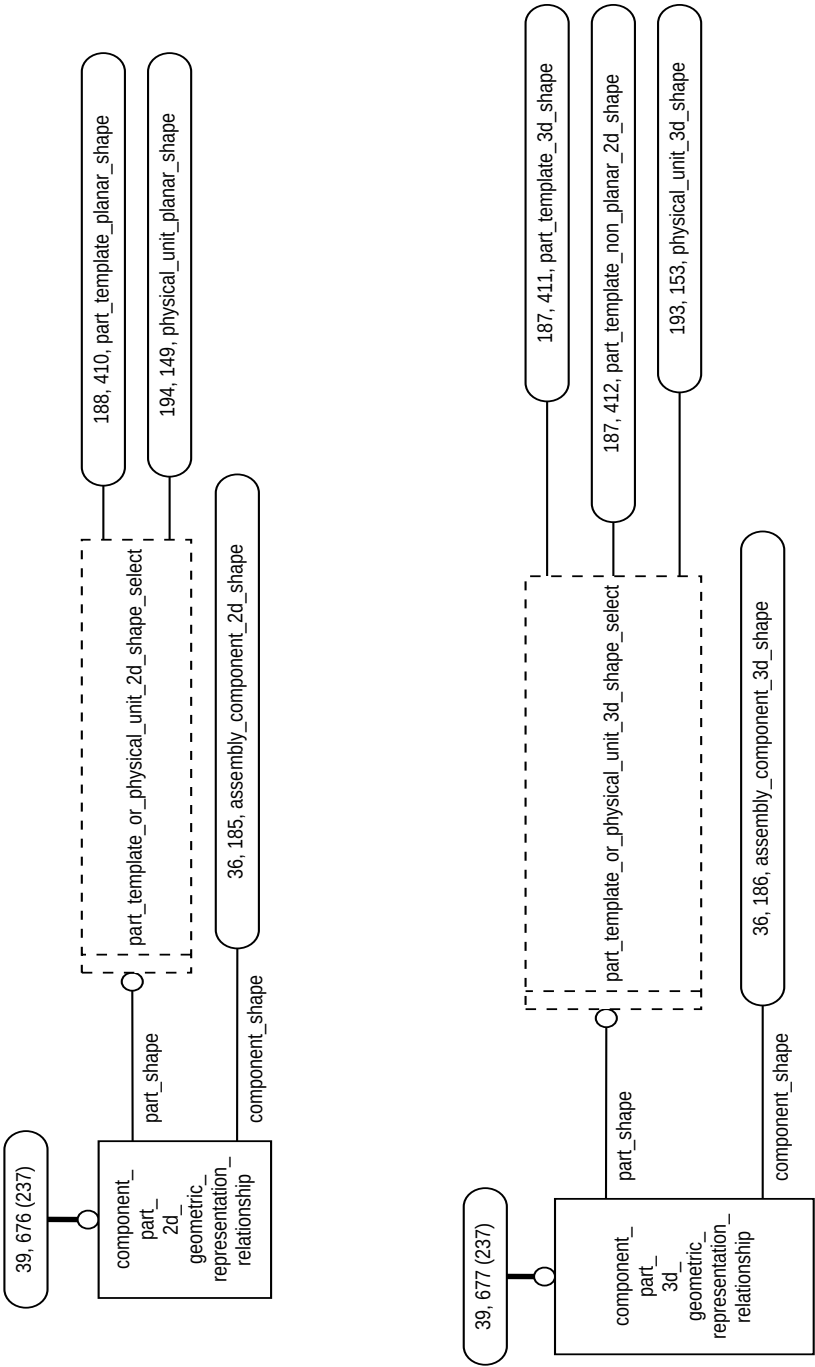


Figure H.39 – ARM EXPRESS-G Graphic 39 of 243

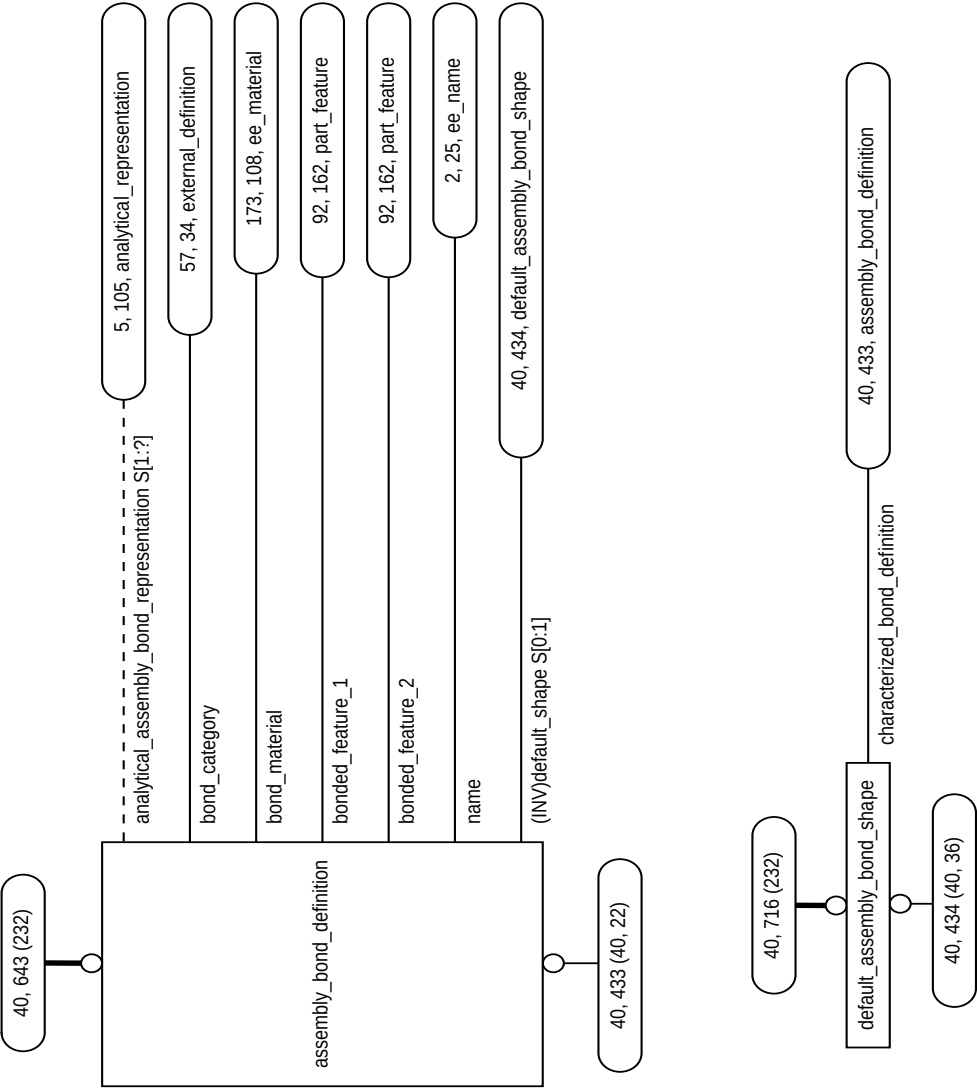


Figure H.40 – ARM EXPRESS-G Graphic 40 of 243

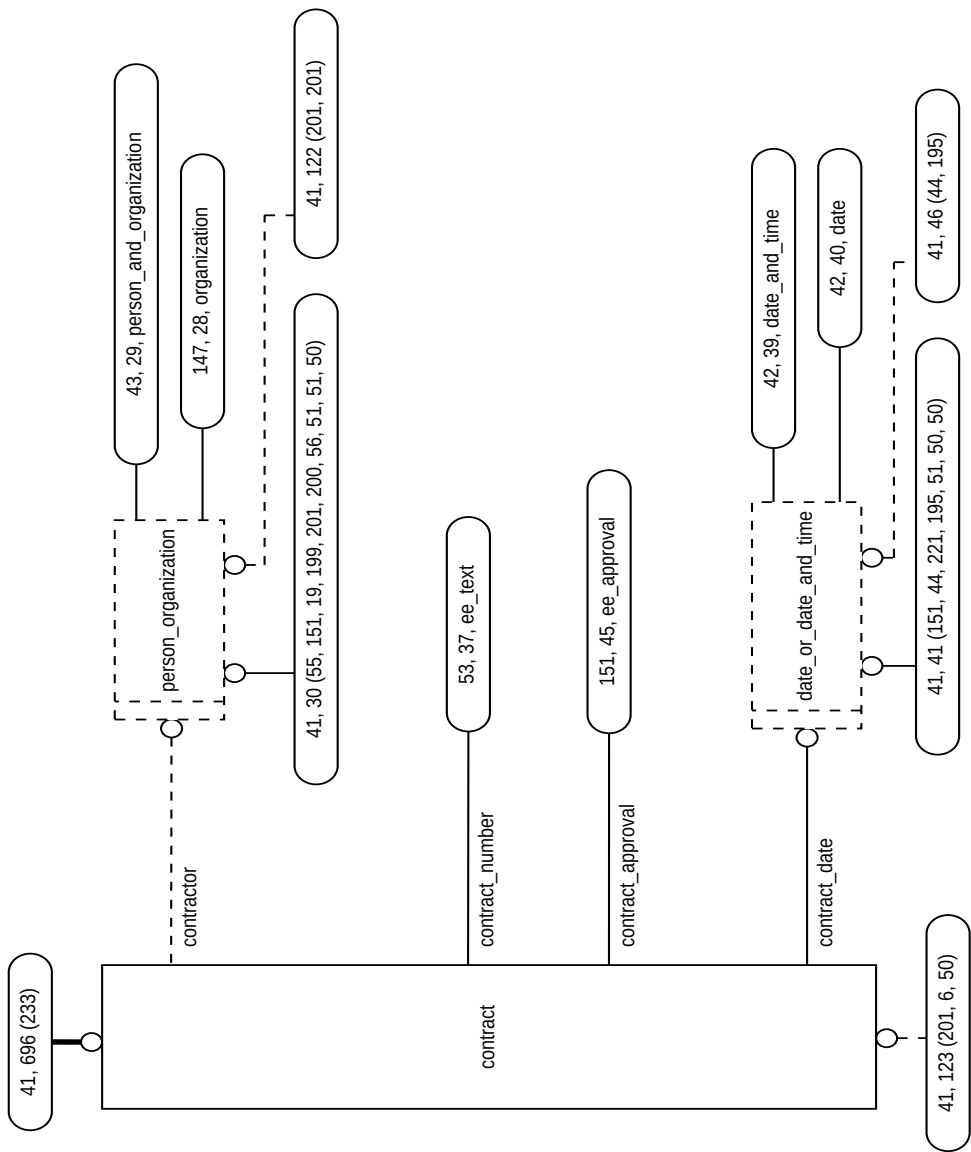


Figure H.41 – ARM EXPRESS-G Graphic 41 of 243

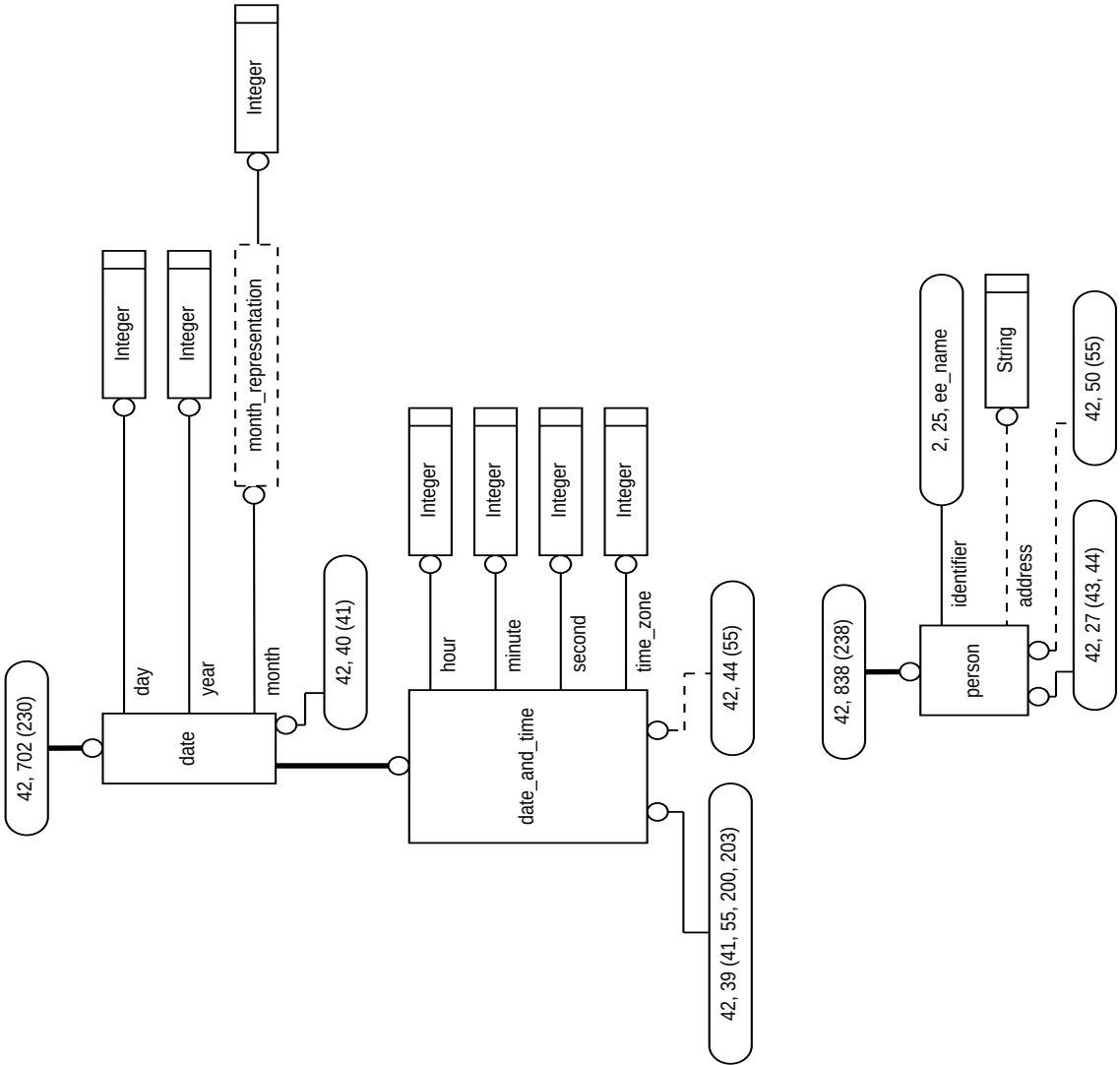


Figure H.42 – ARM EXPRESS-G Graphic 42 of 243

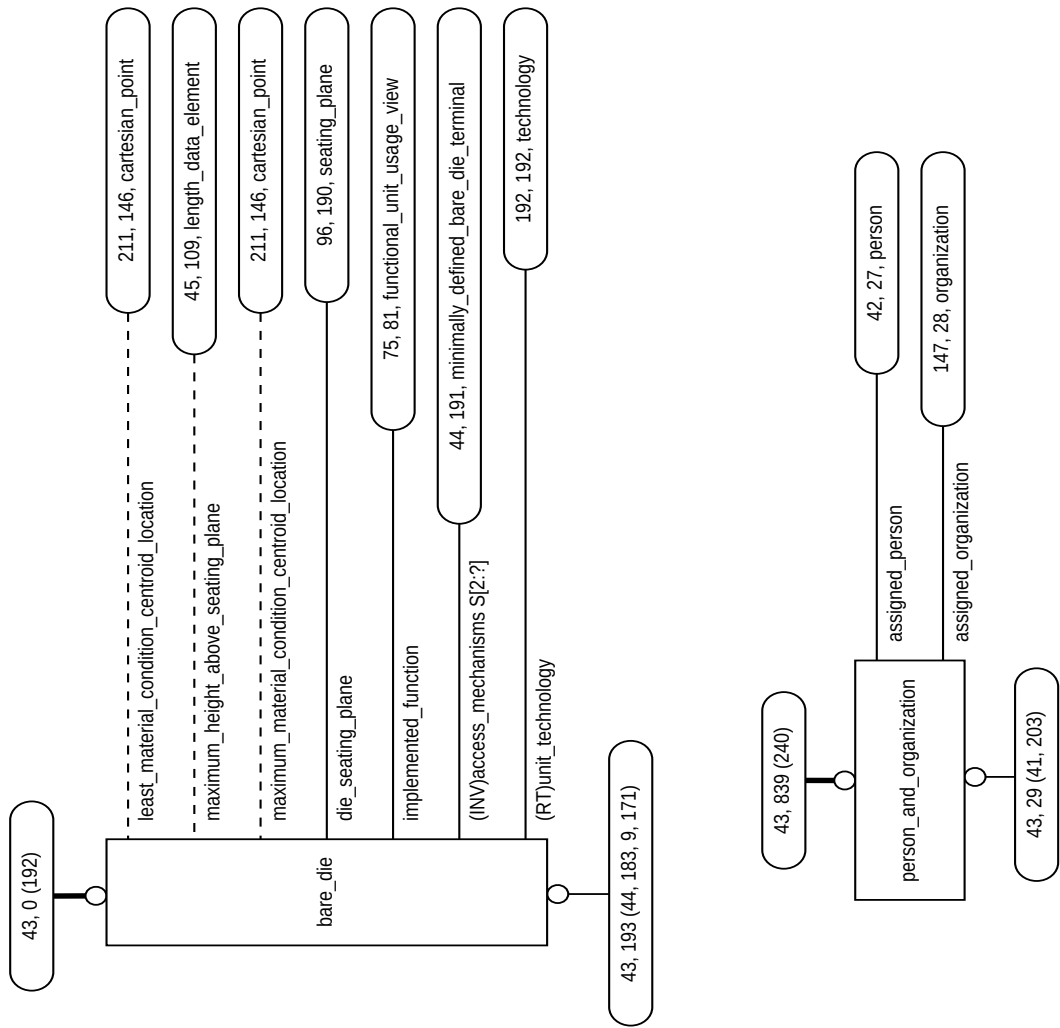


Figure H.43 – ARM EXPRESS-G Graphic 43 of 243

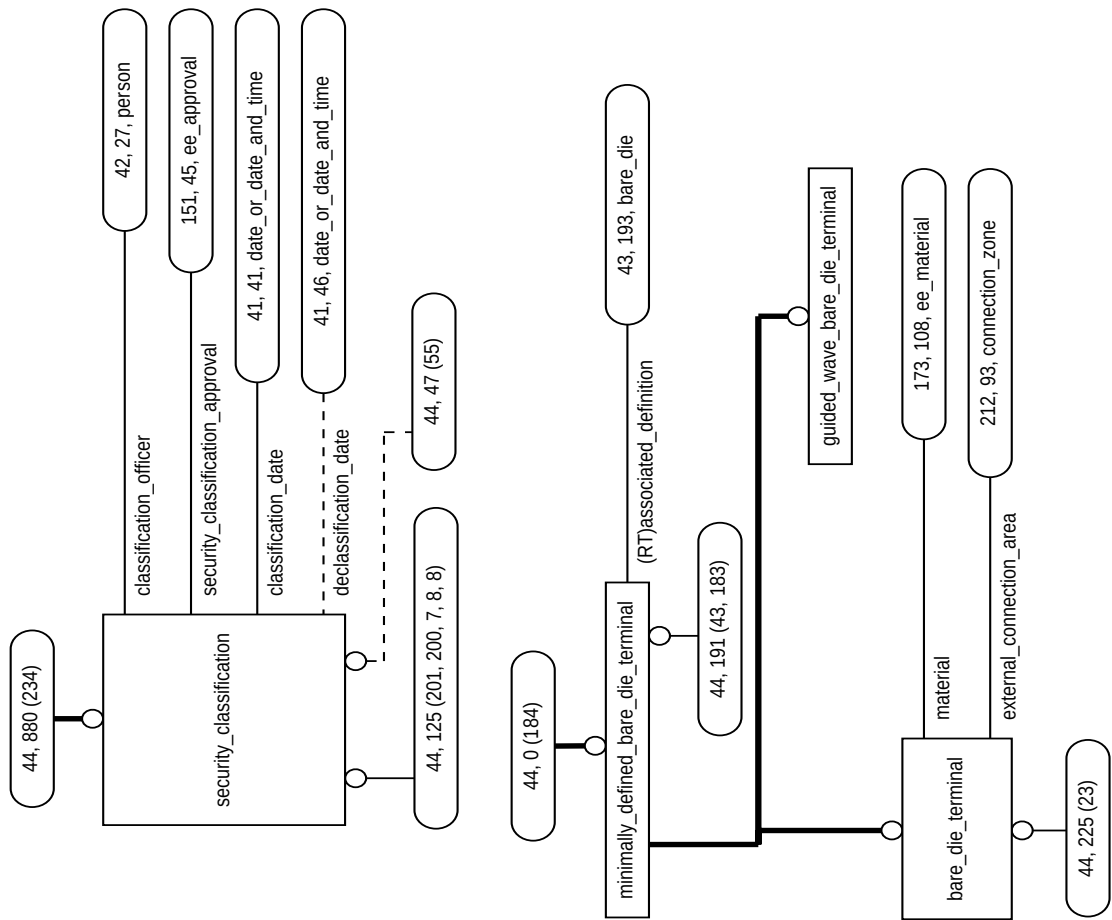


Figure H.44 – ARM EXPRESS-G Graphic 44 of 243

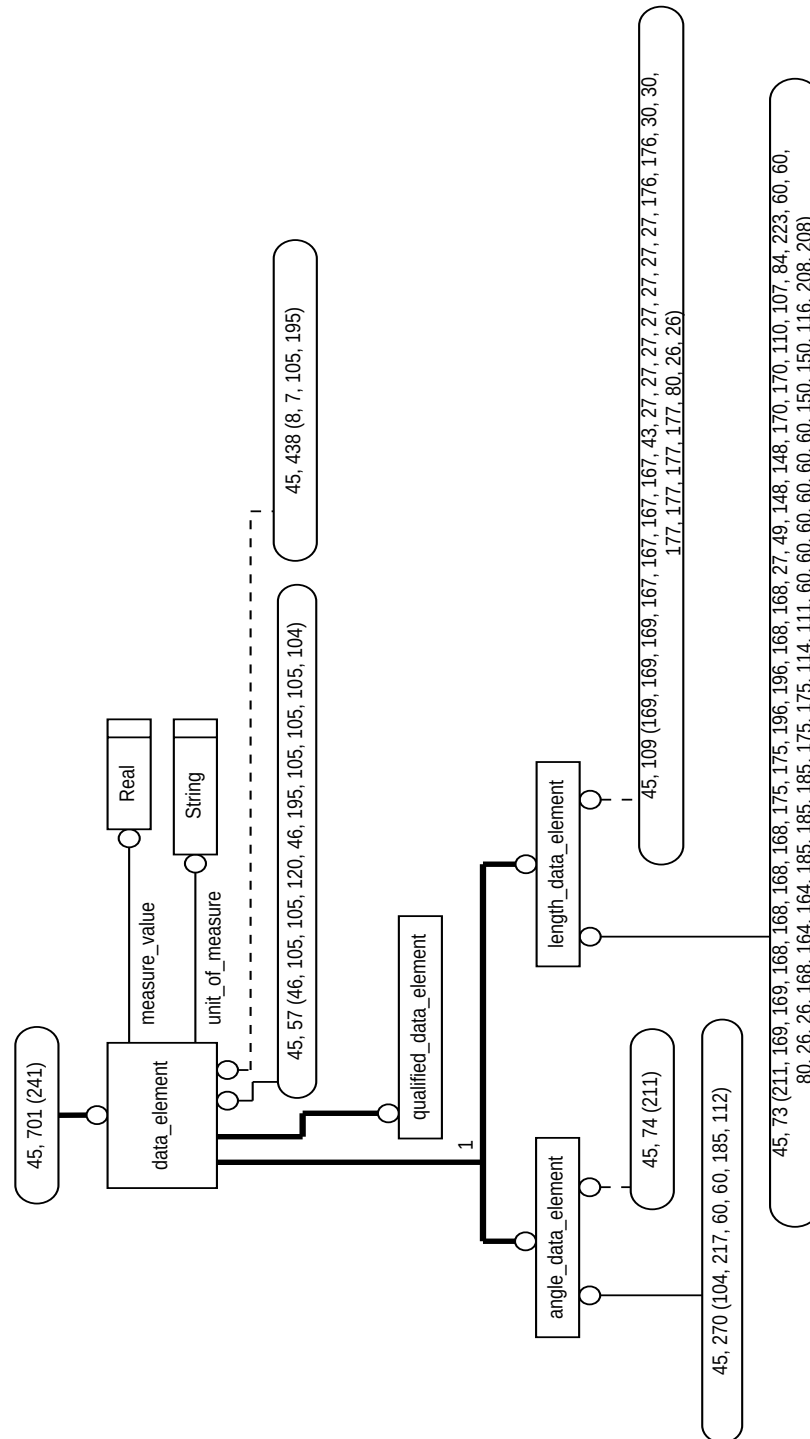


Figure H.45 – ARM EXPRESS-G Graphic 45 of 243

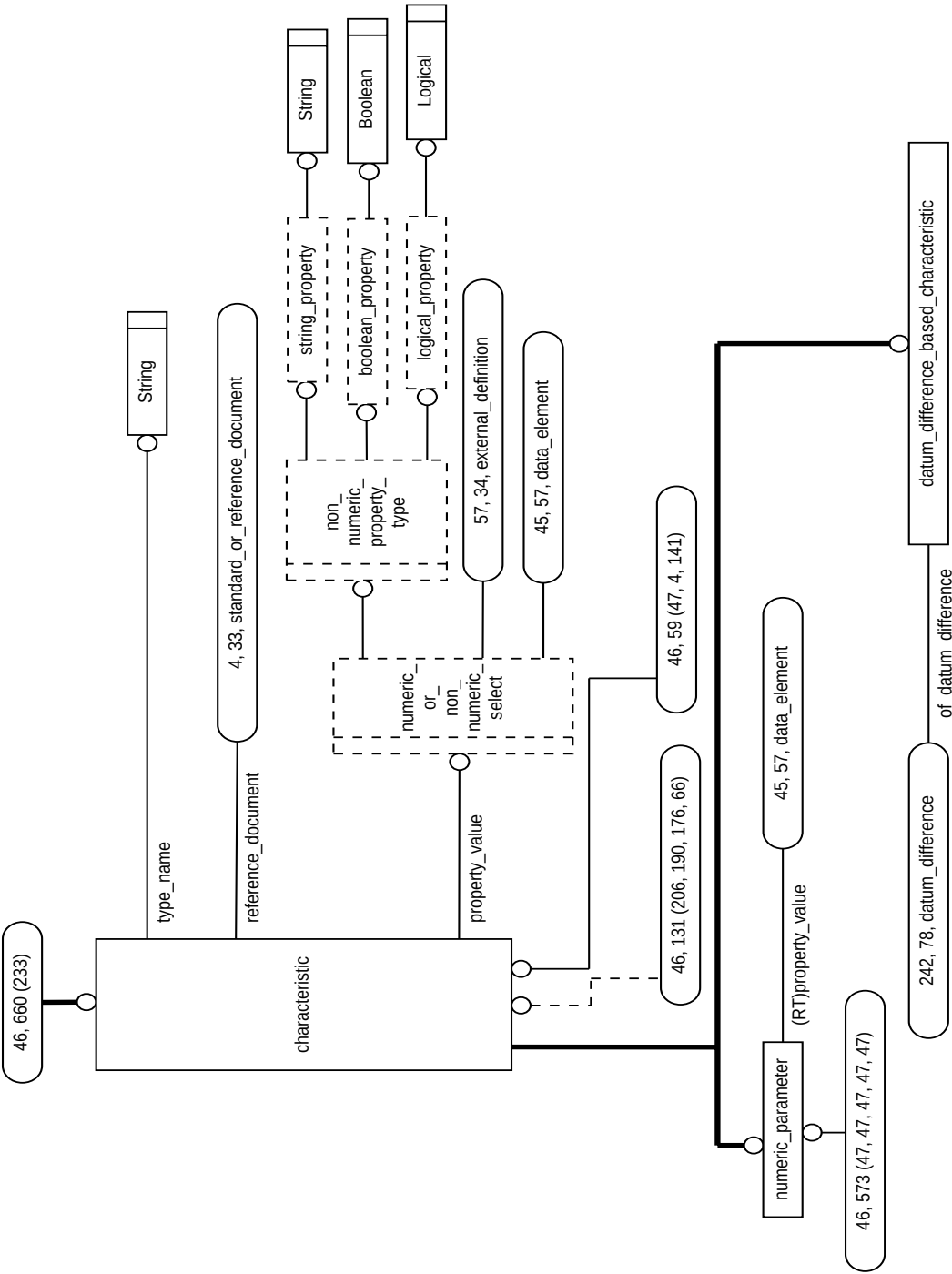


Figure H.46 – ARM EXPRESS-G Graphic 46 of 243

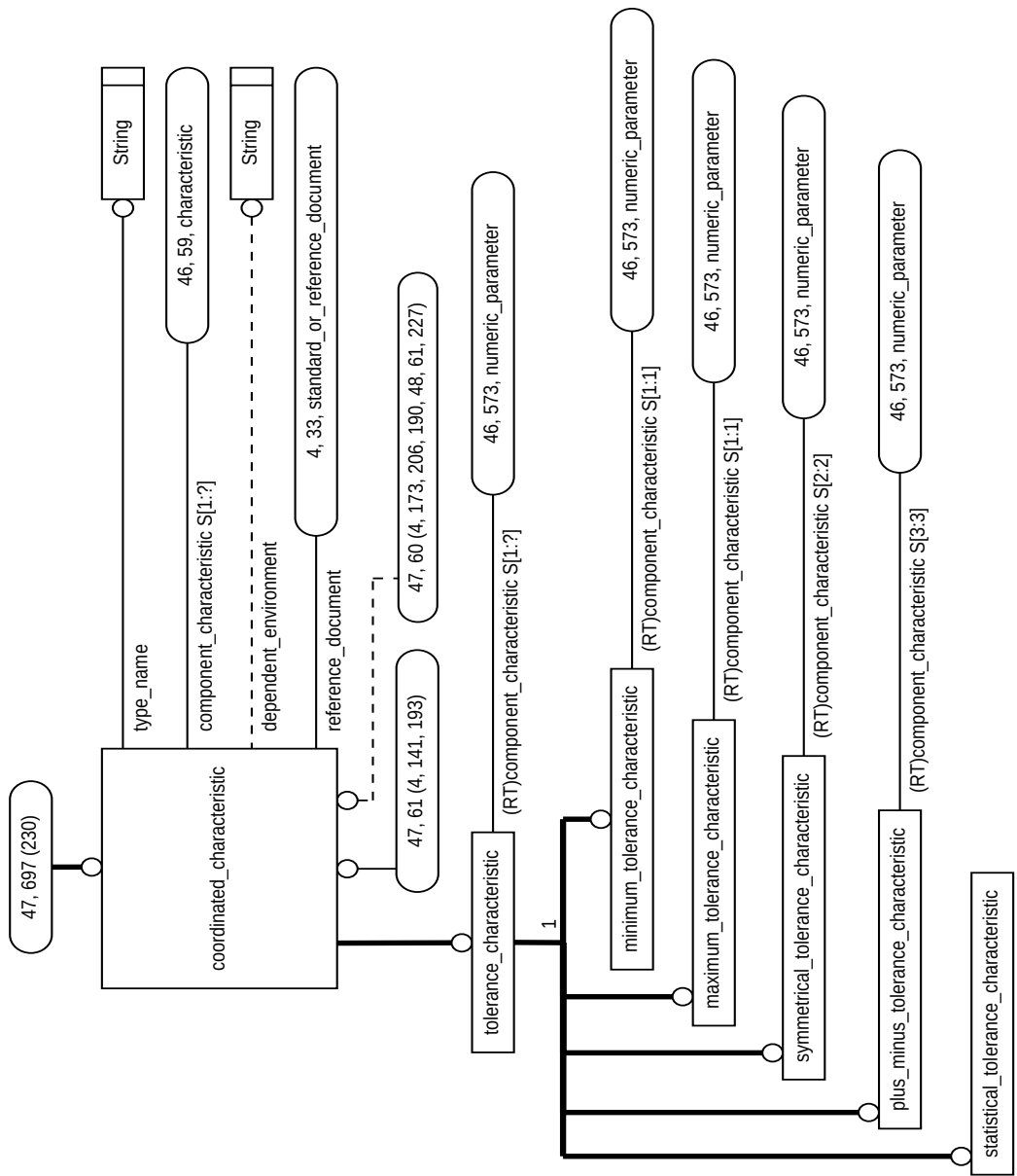


Figure H.47 – ARM EXPRESS-G Graphic 47 of 243

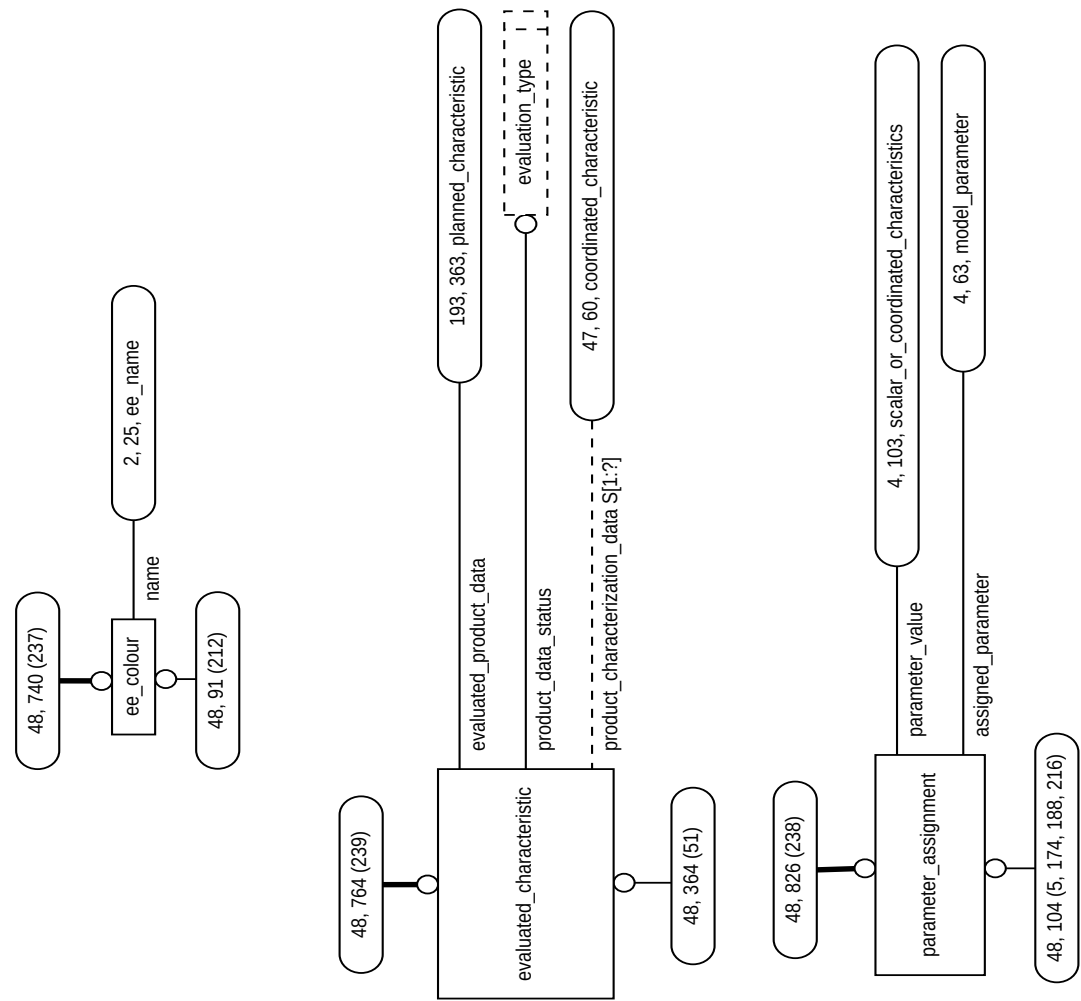


Figure H.48 – ARM EXPRESS-G Graphic 48 of 243

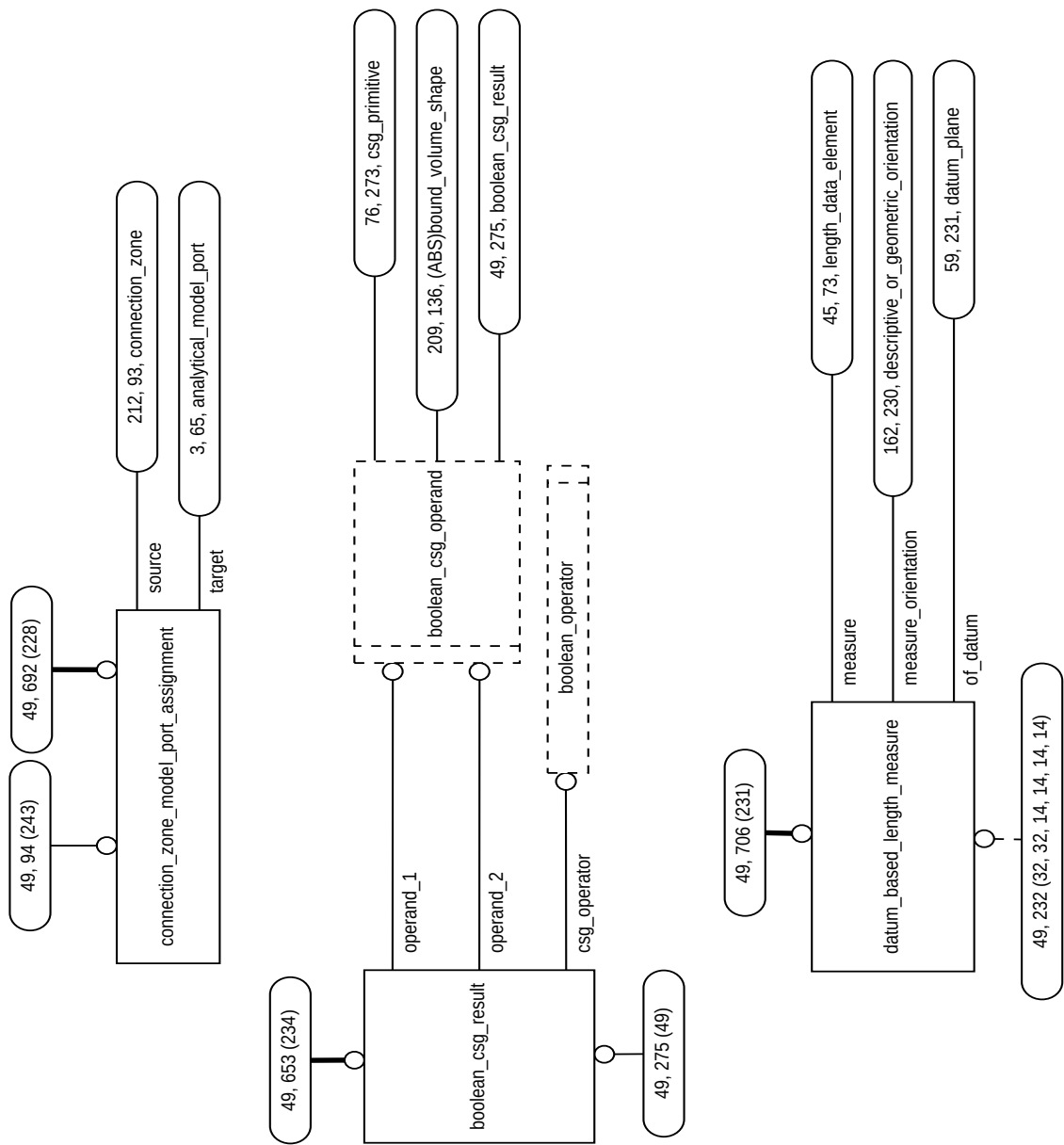


Figure H.49 – ARM EXPRESS-G Graphic 49 of 243

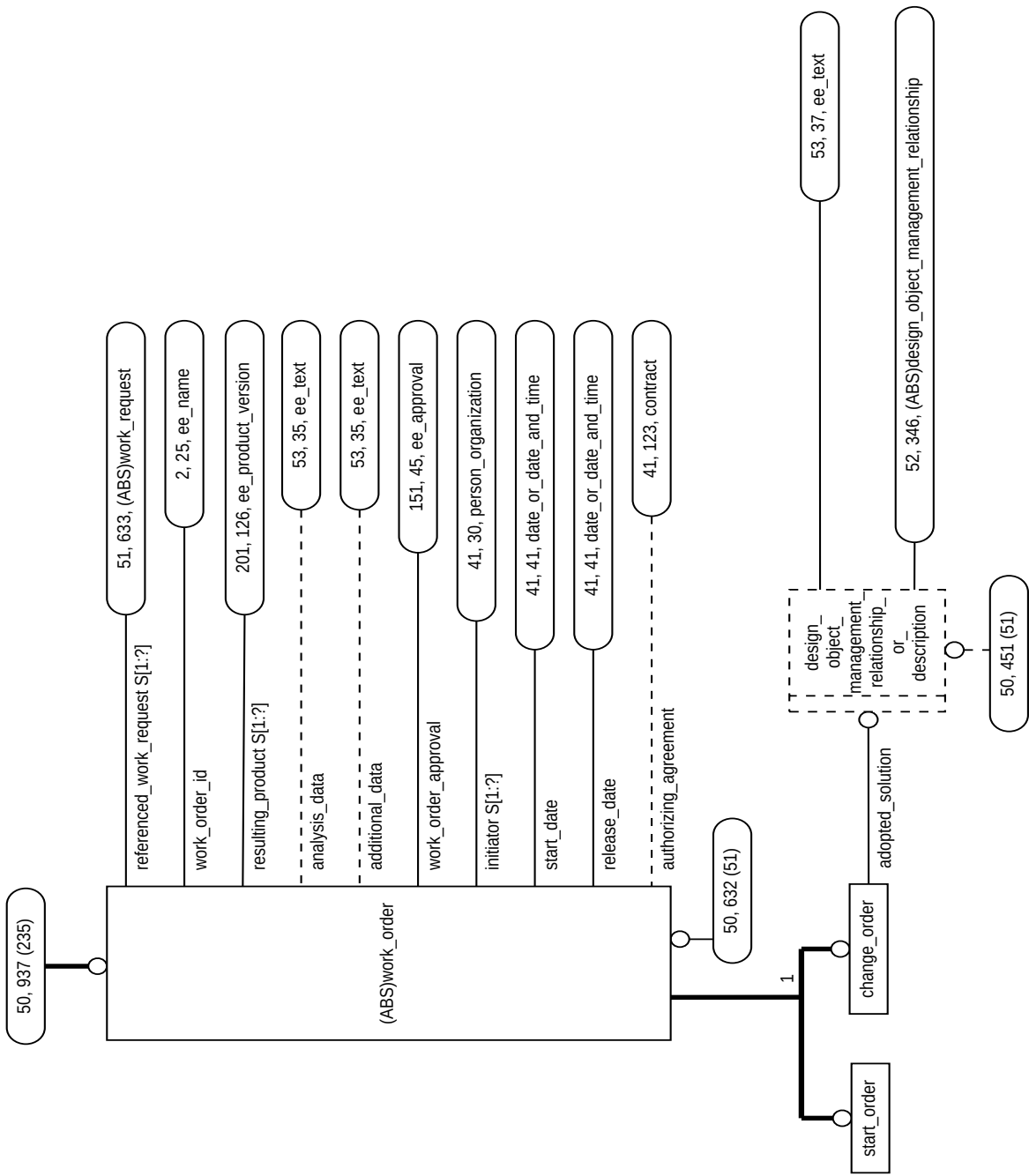


Figure H.50 – ARM EXPRESS-G Graphic 50 of 243

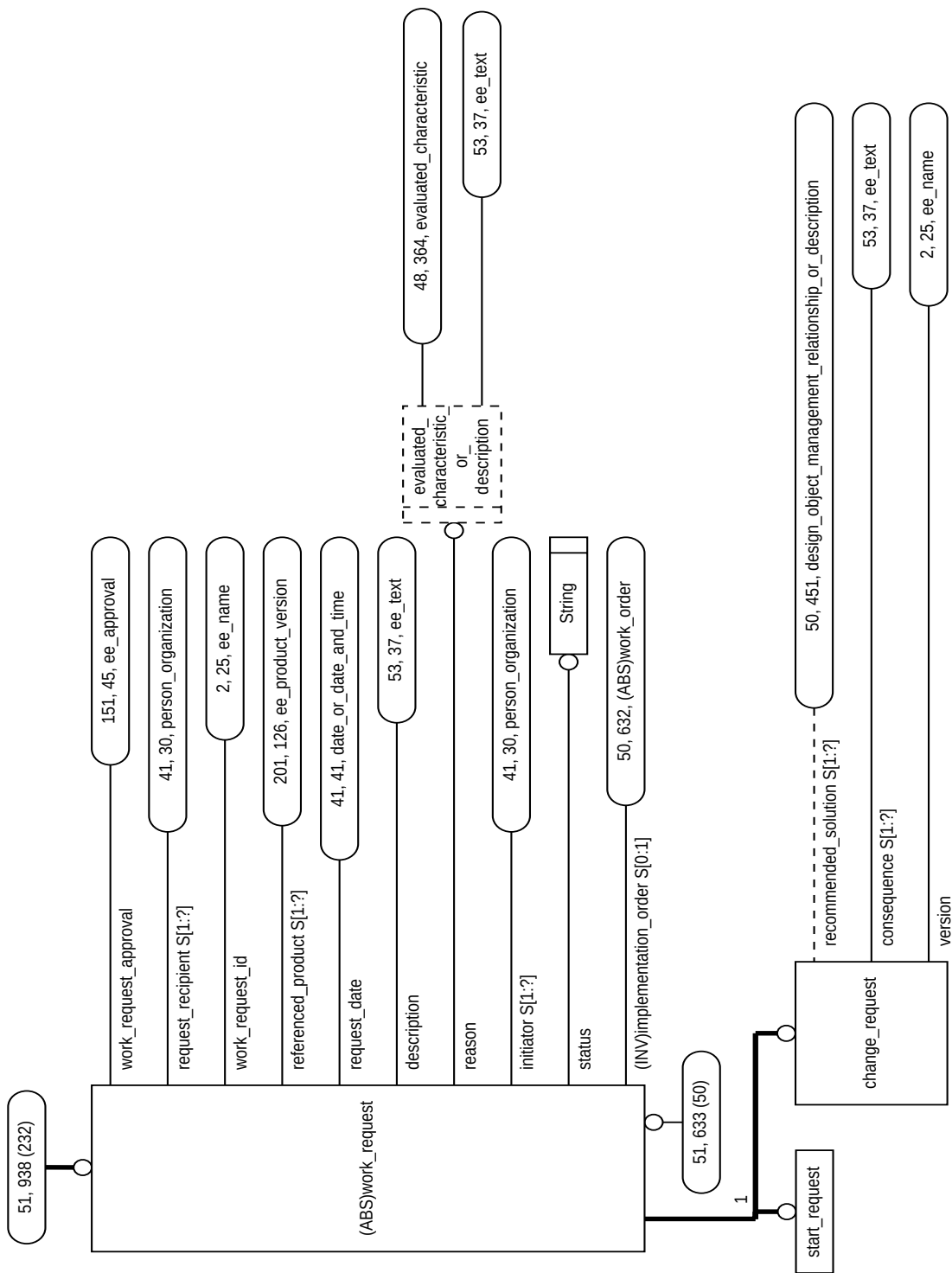


Figure H.51 – ARM EXPRESS-G Graphic 51 of 243

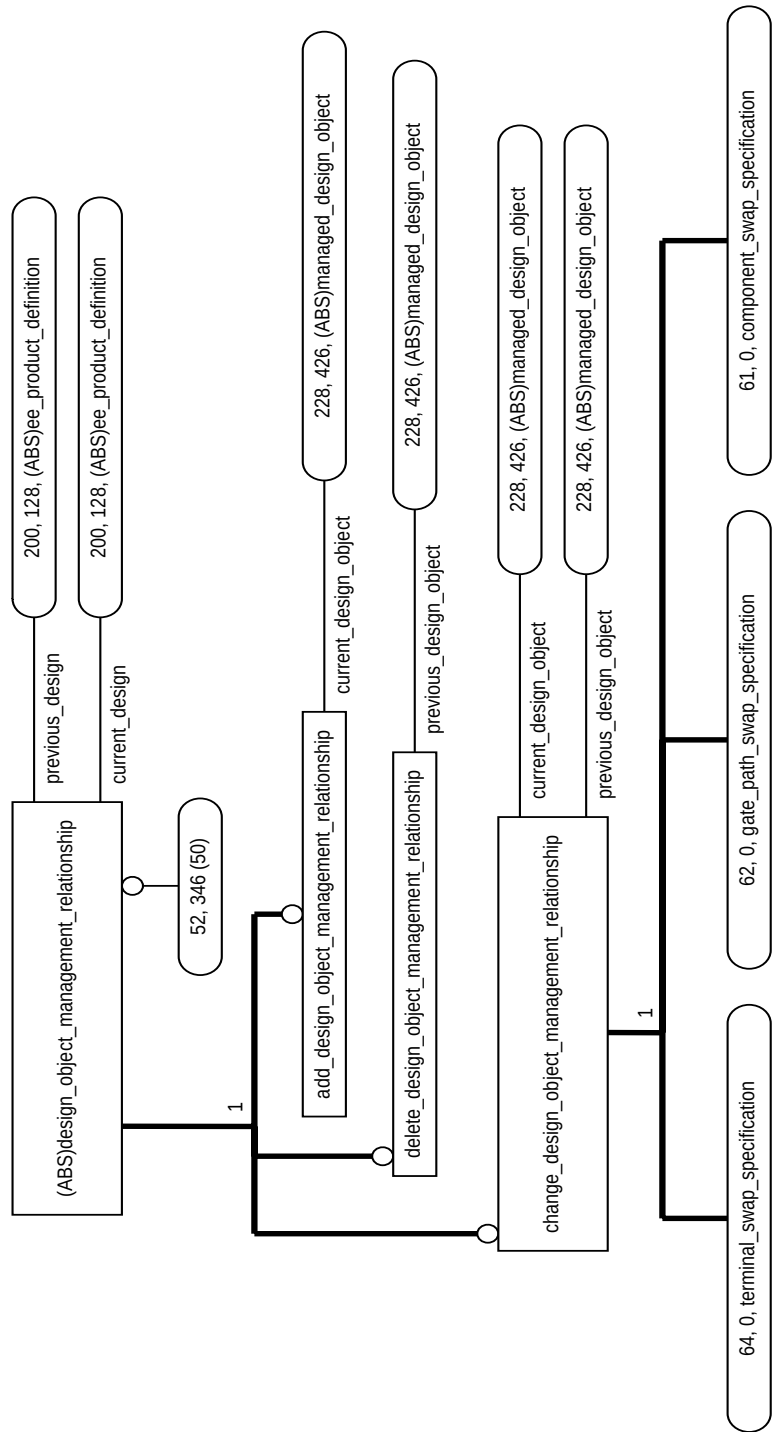


Figure H.52 – ARM EXPRESS-G Graphic 52 of 243

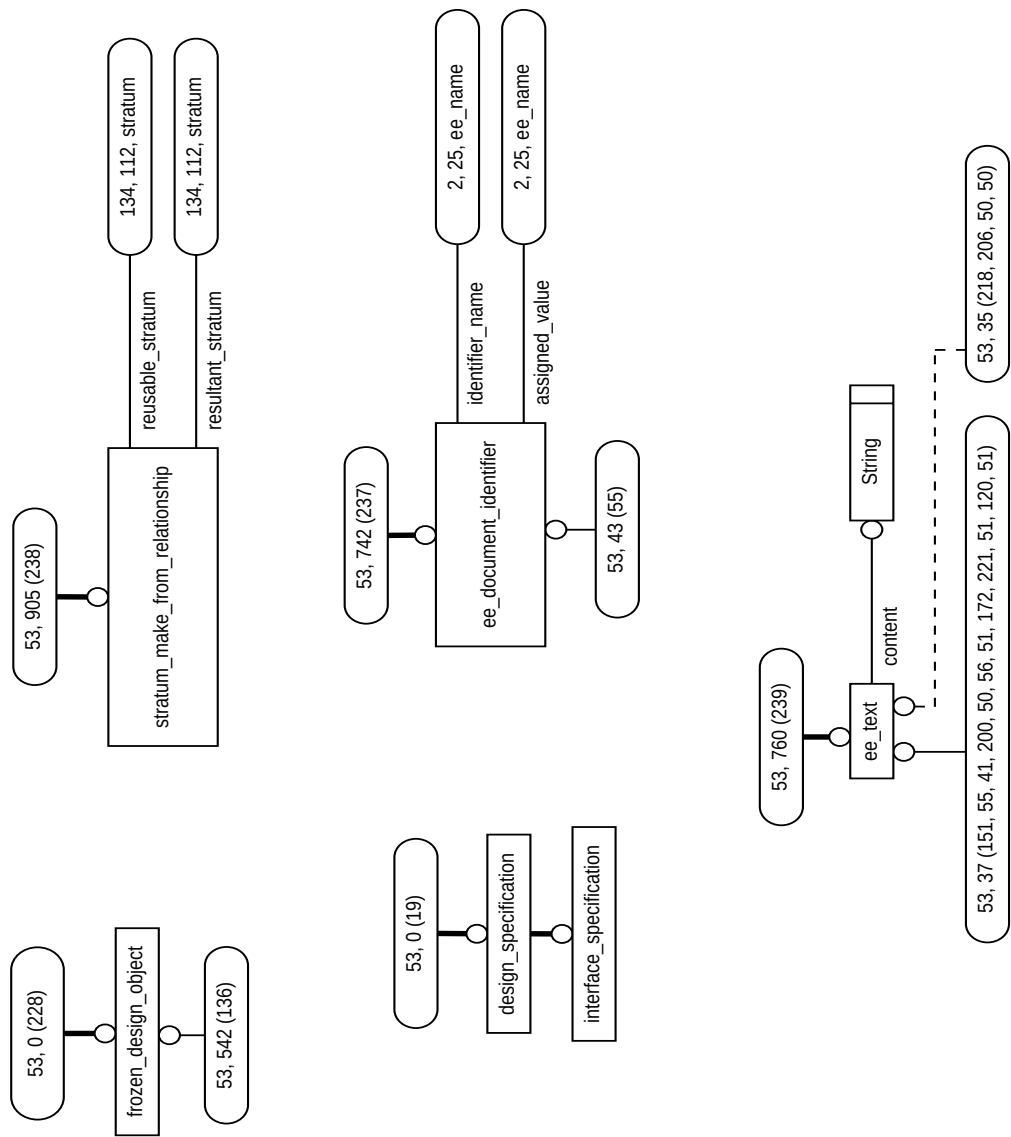


Figure H.53 – ARM EXPRESS-G Graphic 53 of 243

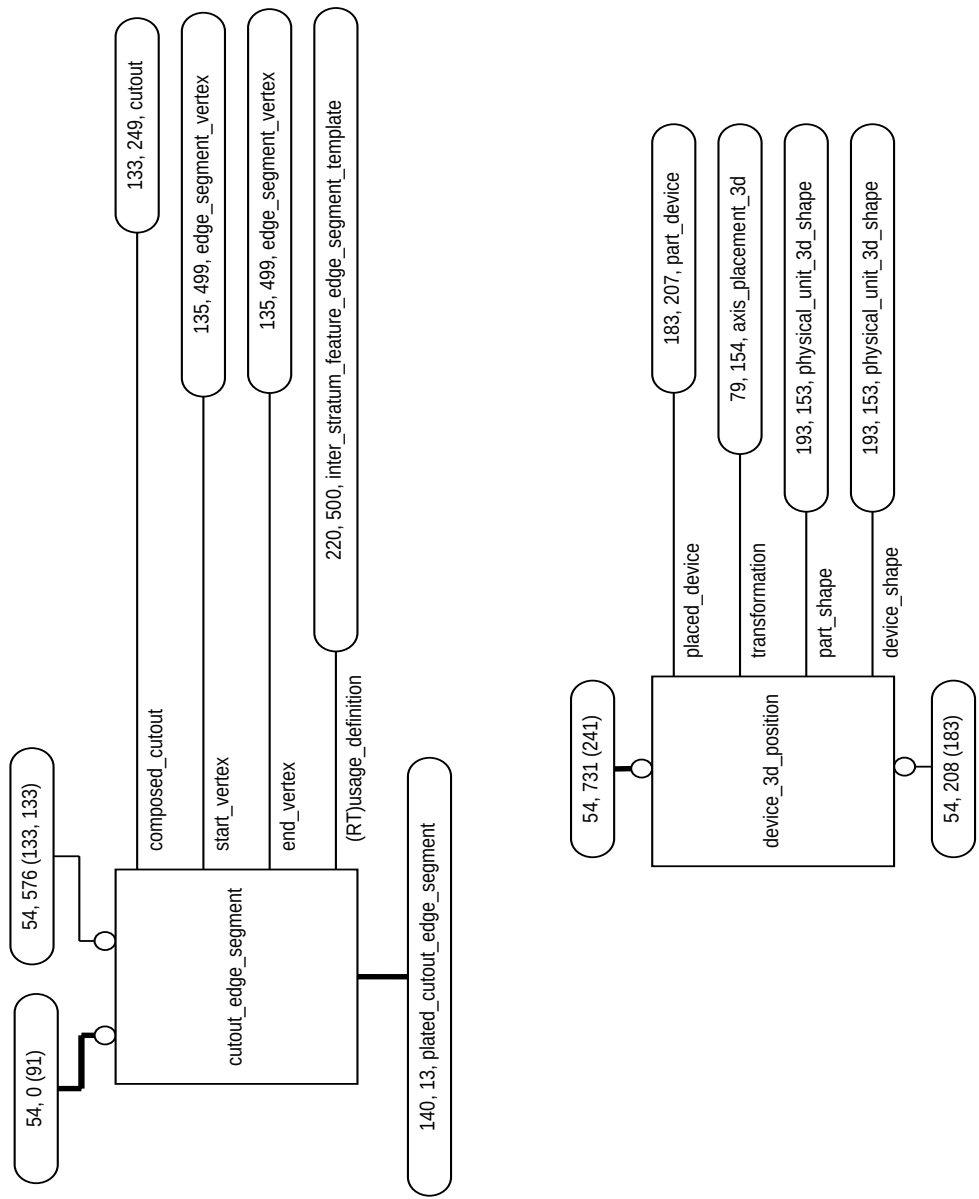


Figure H.54 – ARM EXPRESS-G Graphic 54 of 243

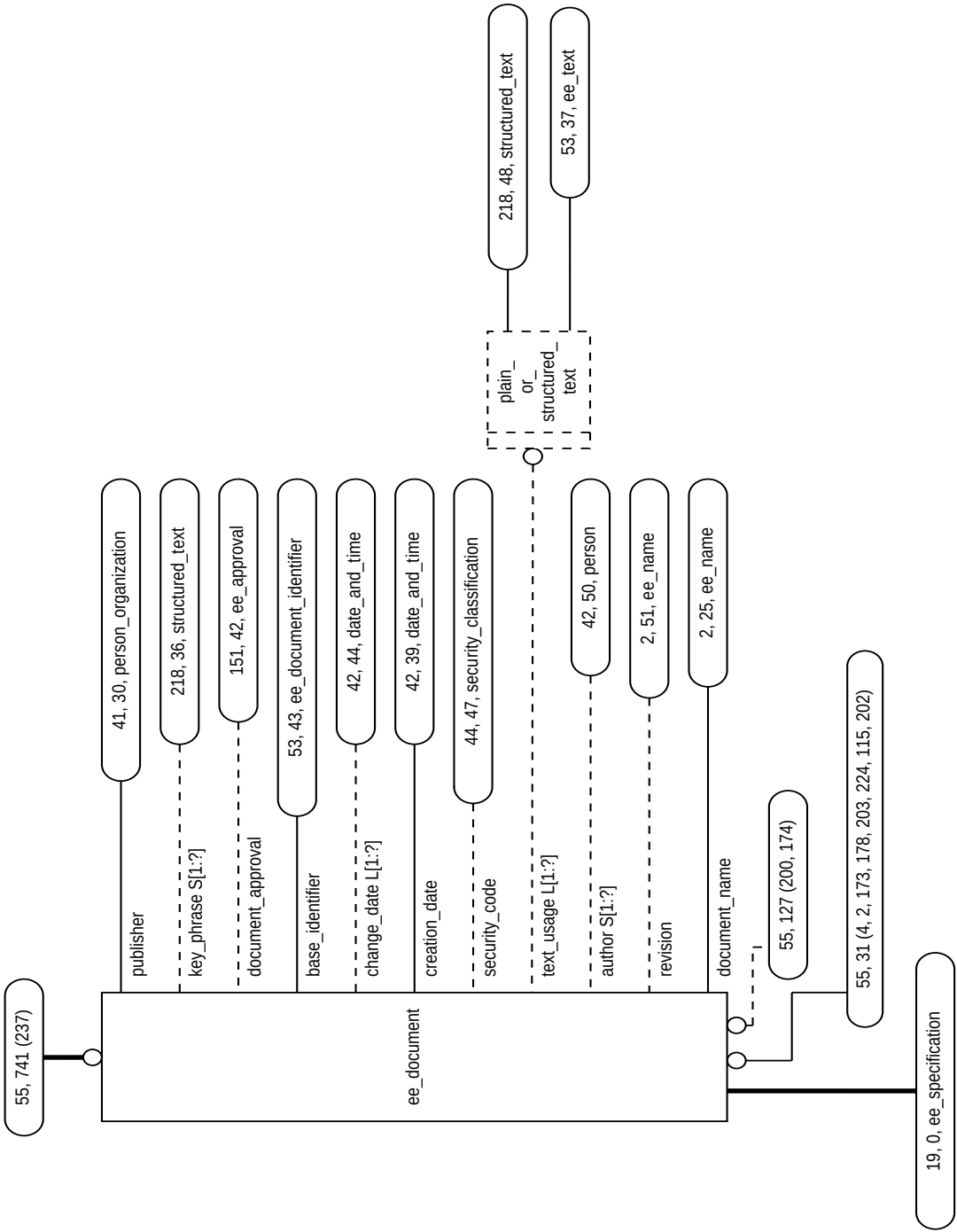


Figure H.55 – ARM EXPRESS-G Graphic 55 of 243

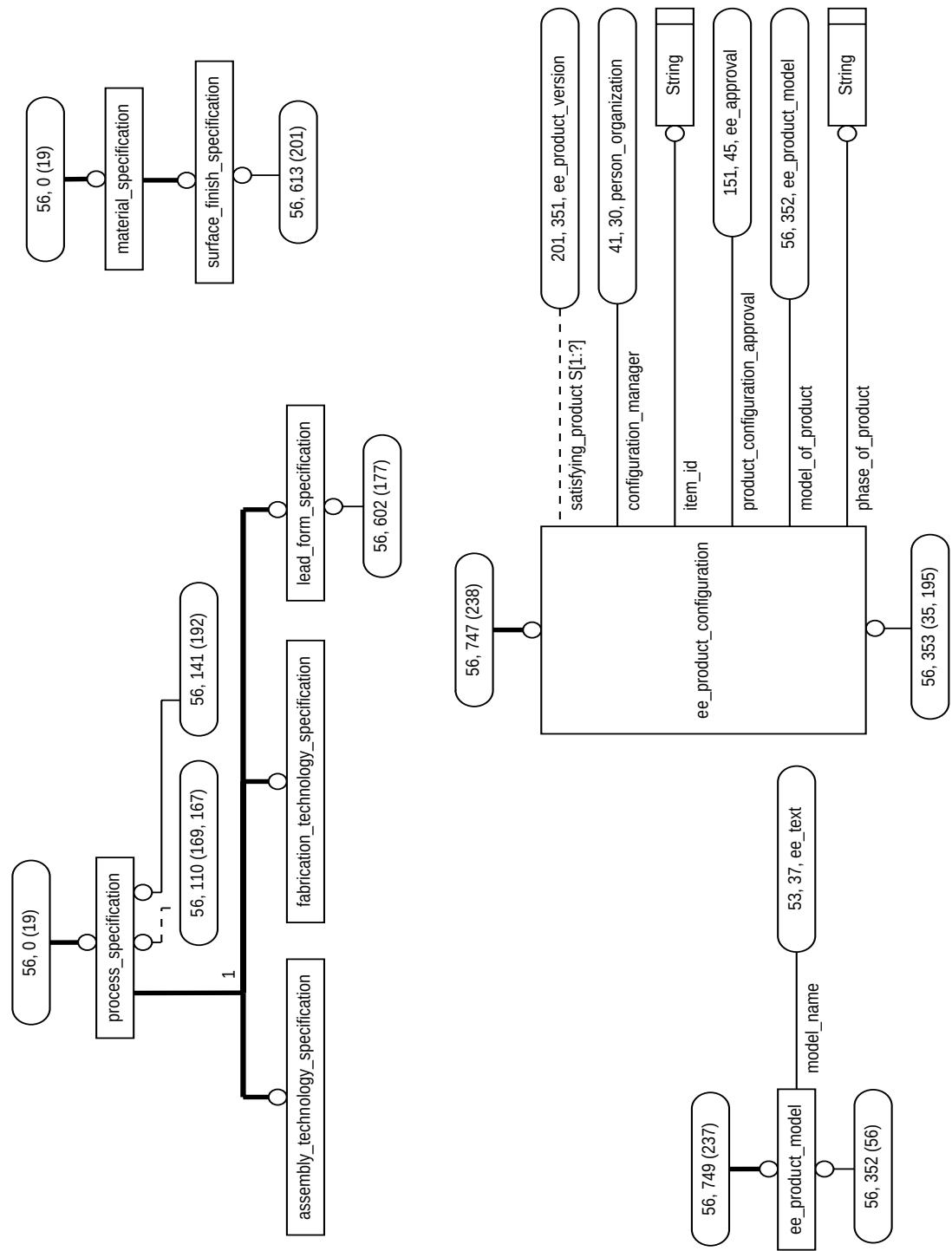


Figure H.56 – ARM EXPRESS-G Graphic 56 of 243

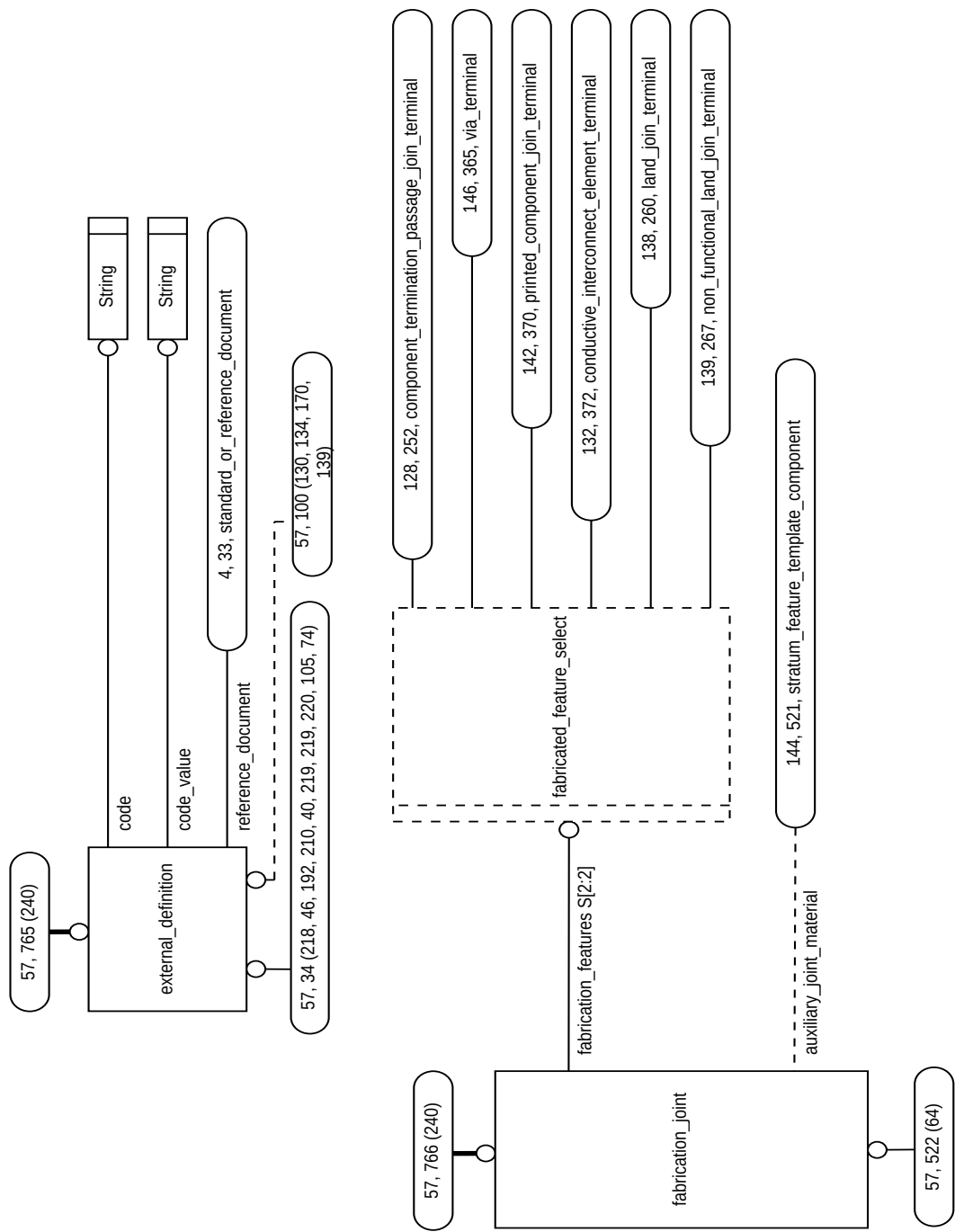


Figure H.57 – ARM EXPRESS-G Graphic 57 of 243

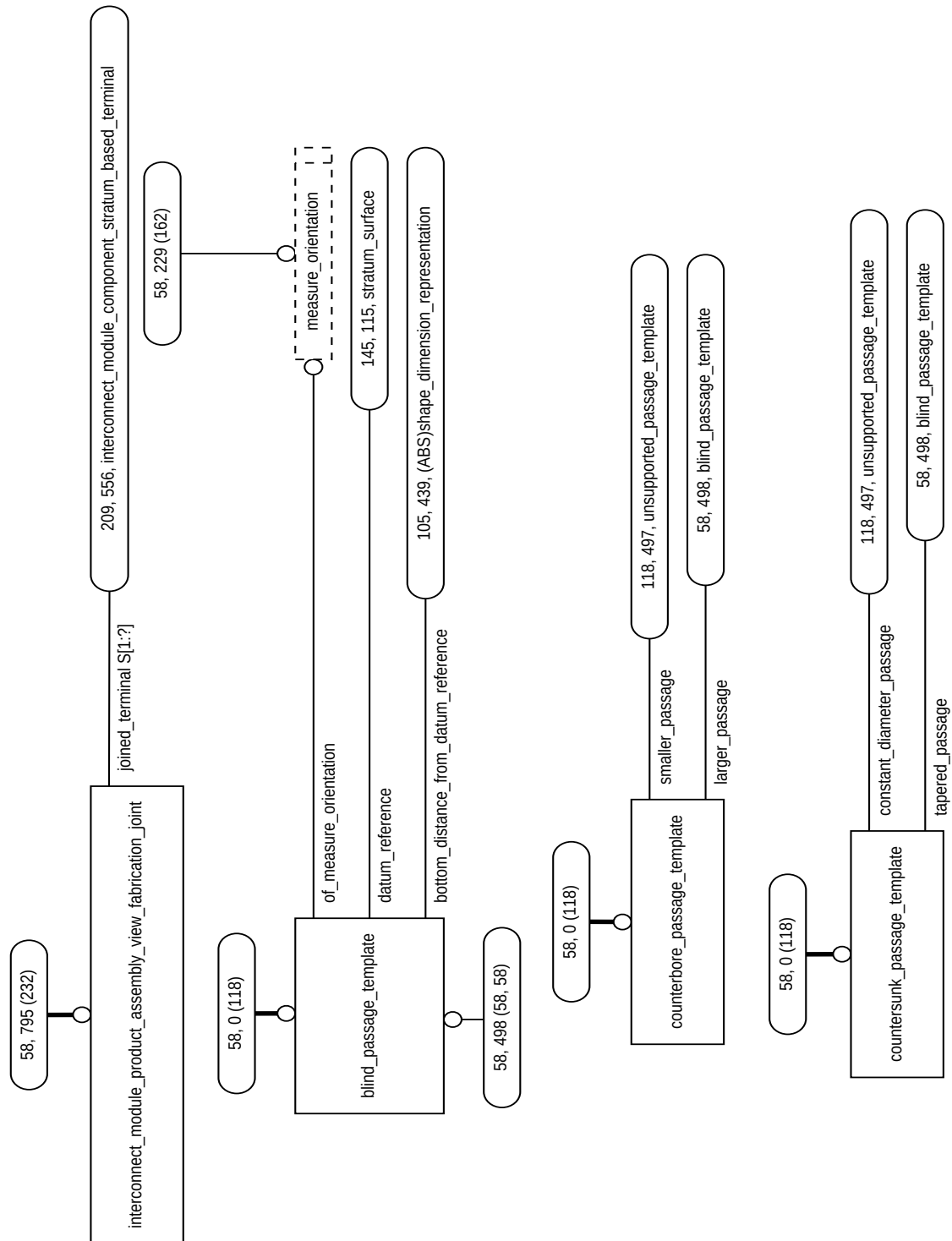


Figure H.58 – ARM EXPRESS-G Graphic 58 of 243

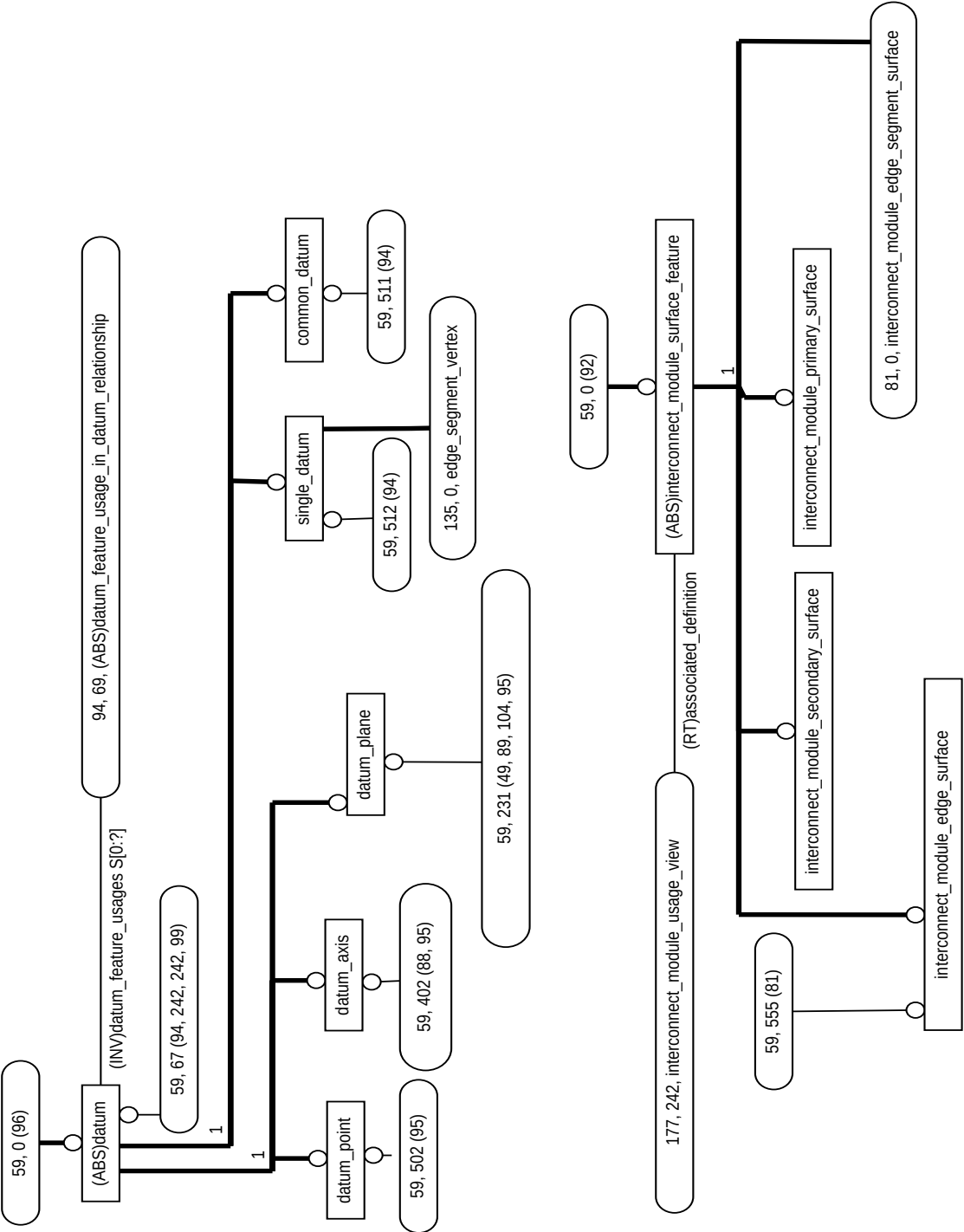


Figure H.59 – ARM EXPRESS-G Graphic 59 of 243

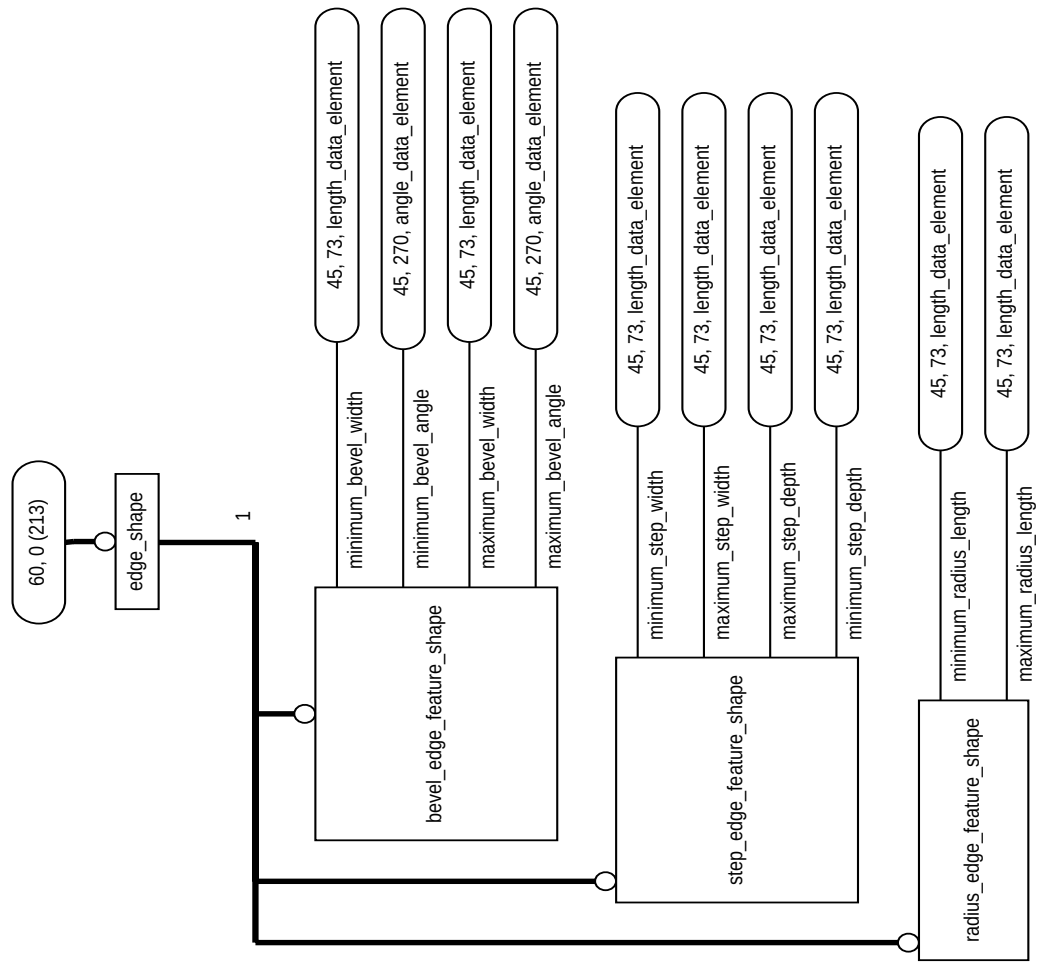


Figure H.60 – ARM EXPRESS-G Graphic 60 of 243

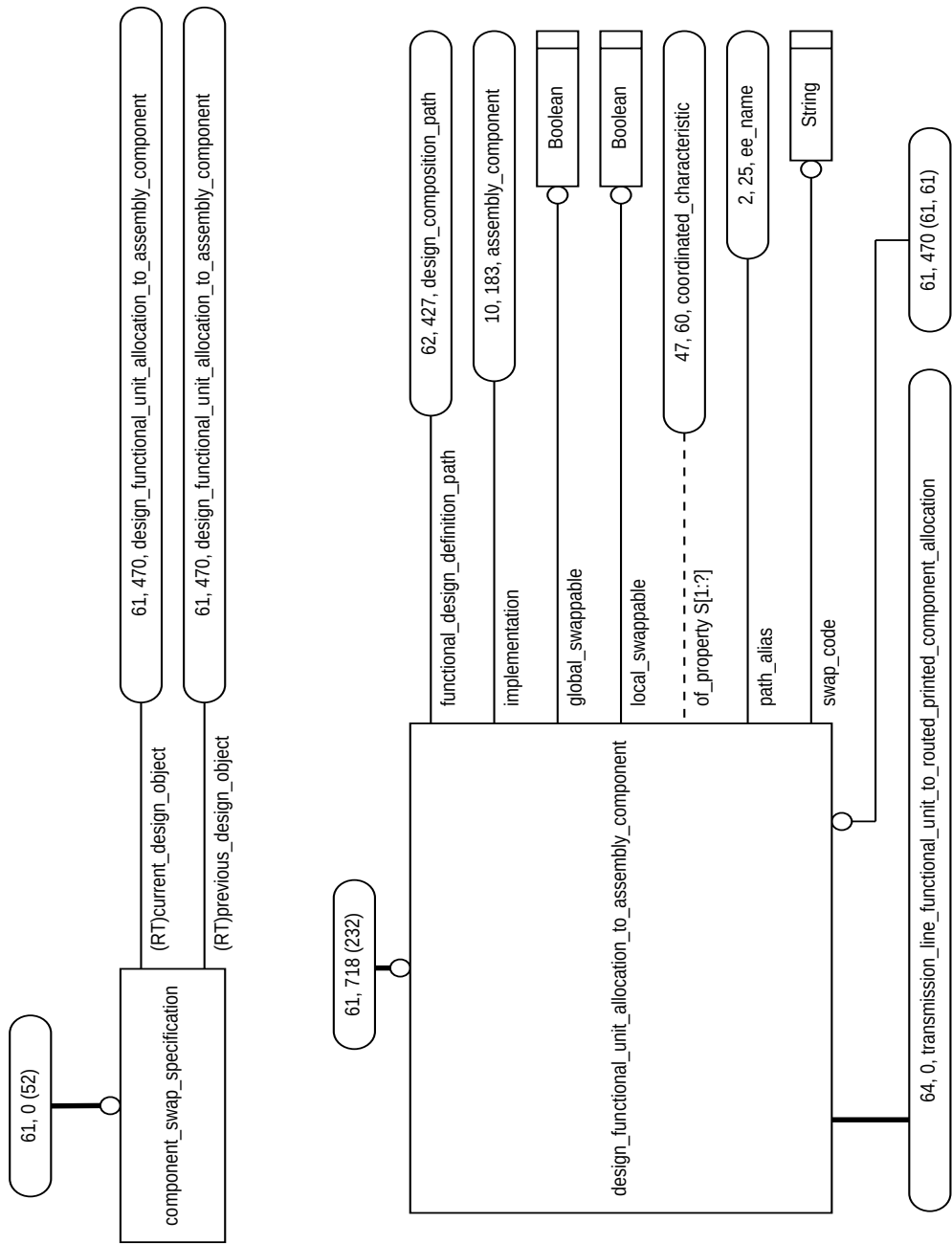


Figure H.61 – ARM EXPRESS-G Graphic 61 of 243

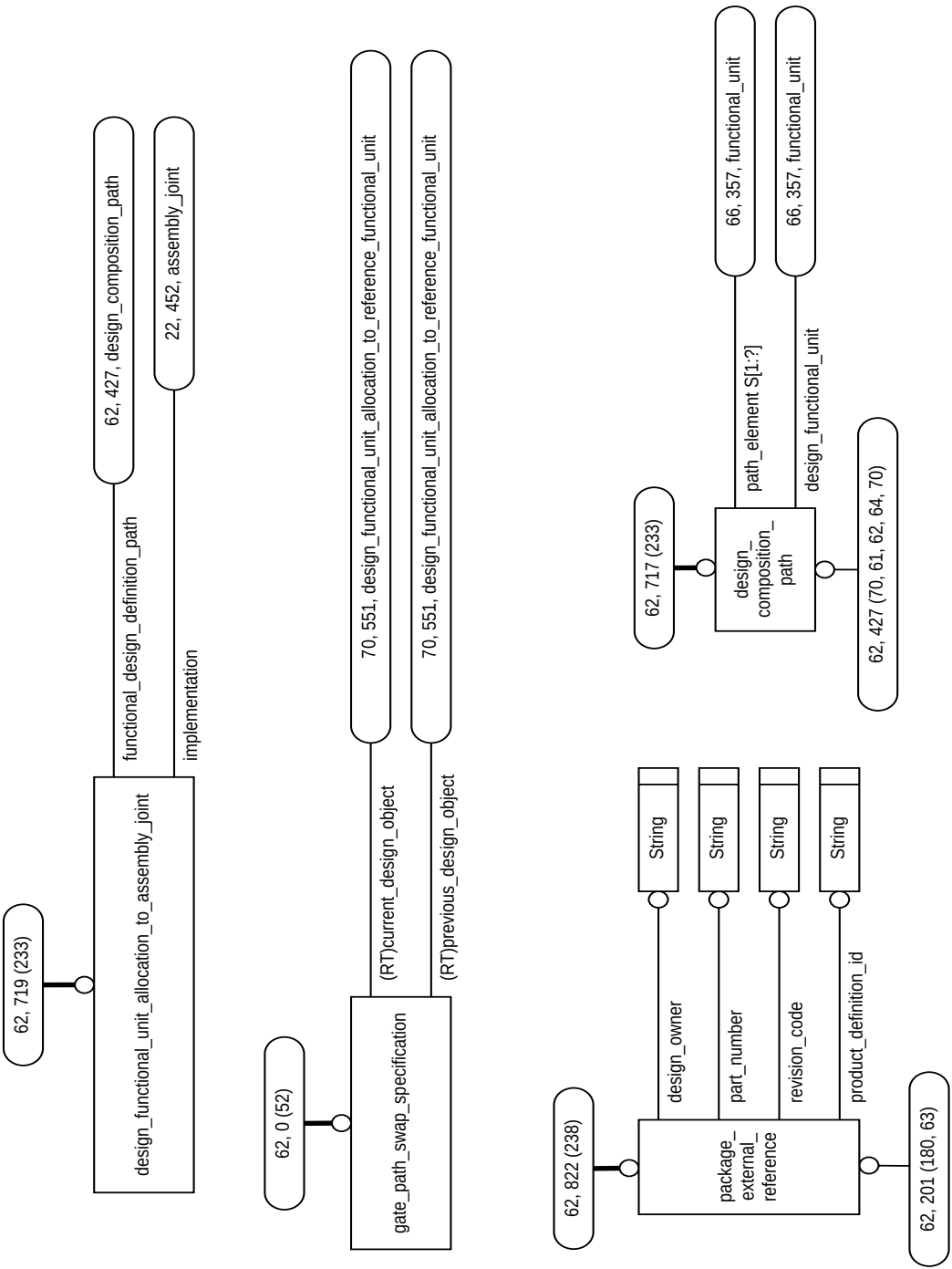


Figure H.62 – ARM EXPRESS-G Graphic 62 of 243

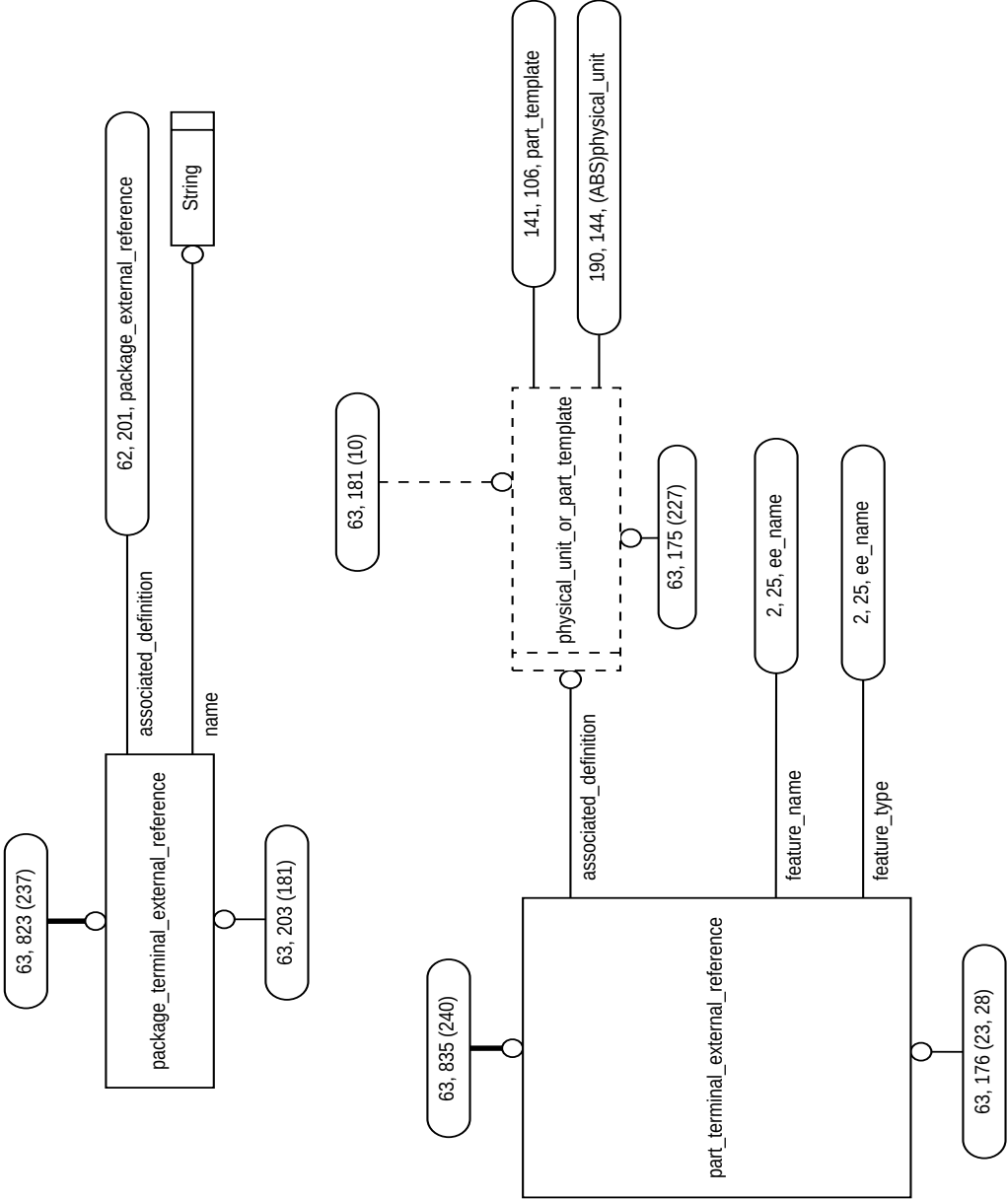


Figure H.63 – ARM EXPRESS-G Graphic 63 of 243

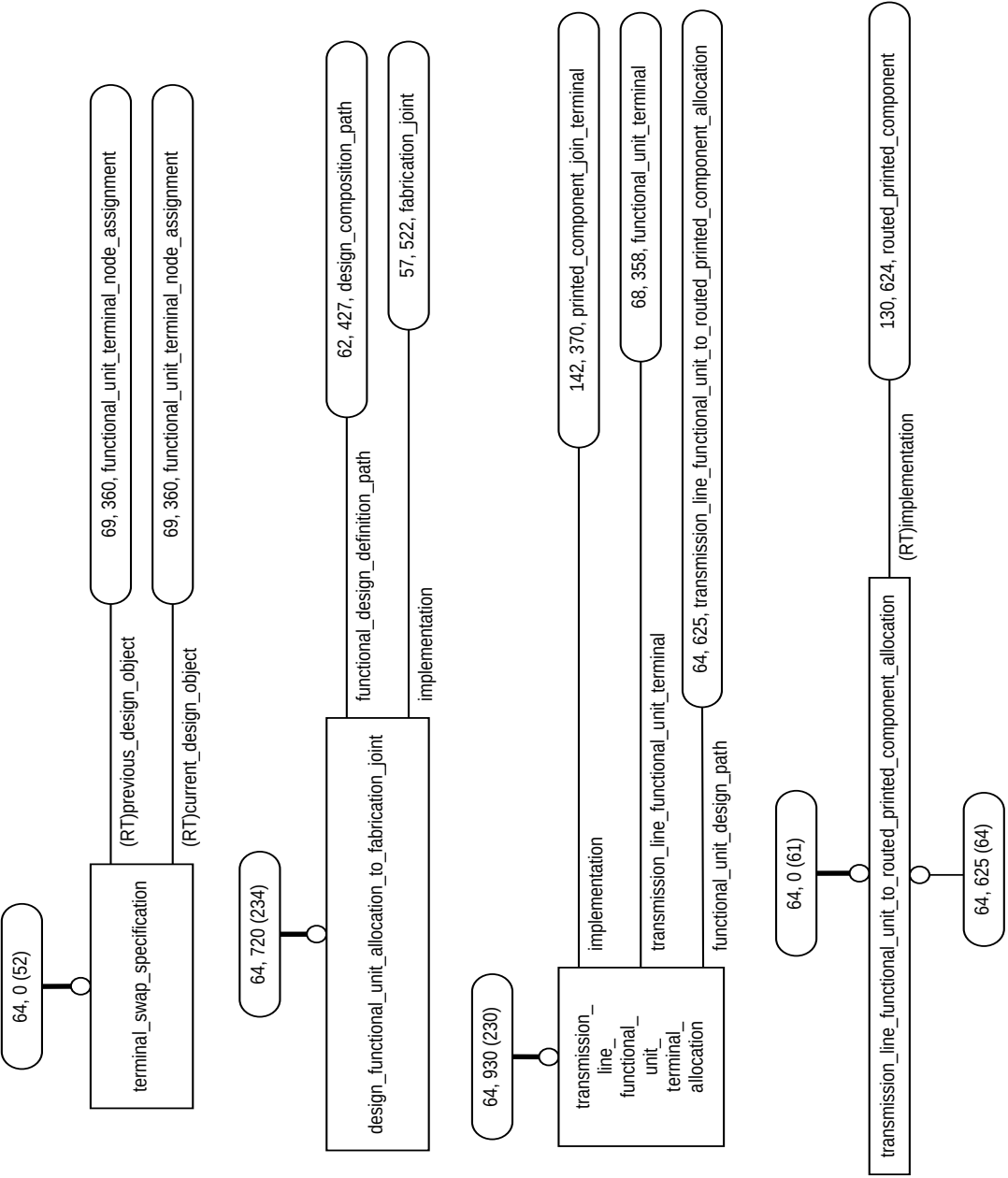


Figure H.64 – ARM EXPRESS-G Graphic 64 of 243

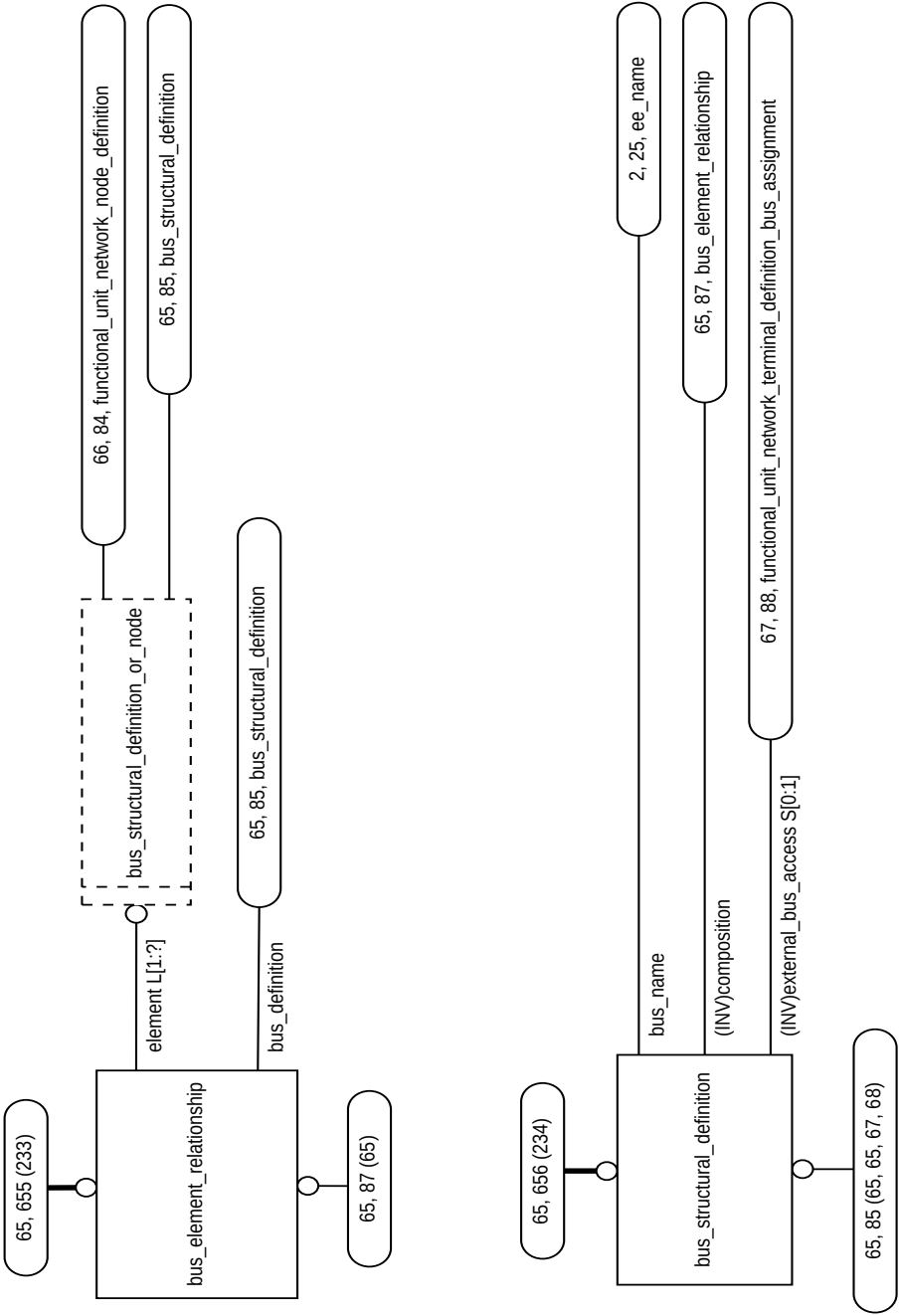


Figure H.65 – ARM EXPRESS-G Graphic 65 of 243

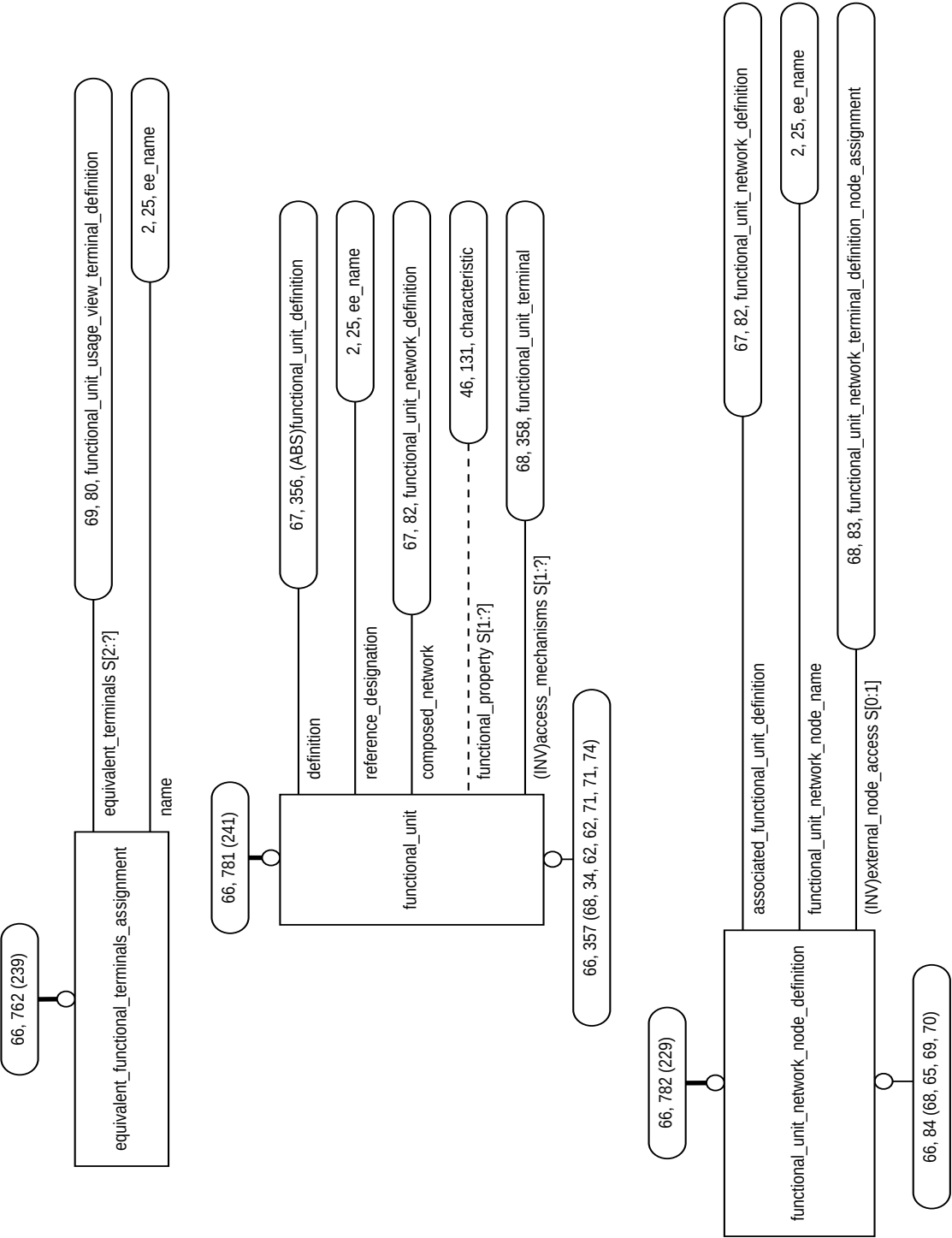


Figure H.66 – ARM EXPRESS-G Graphic 66 of 243

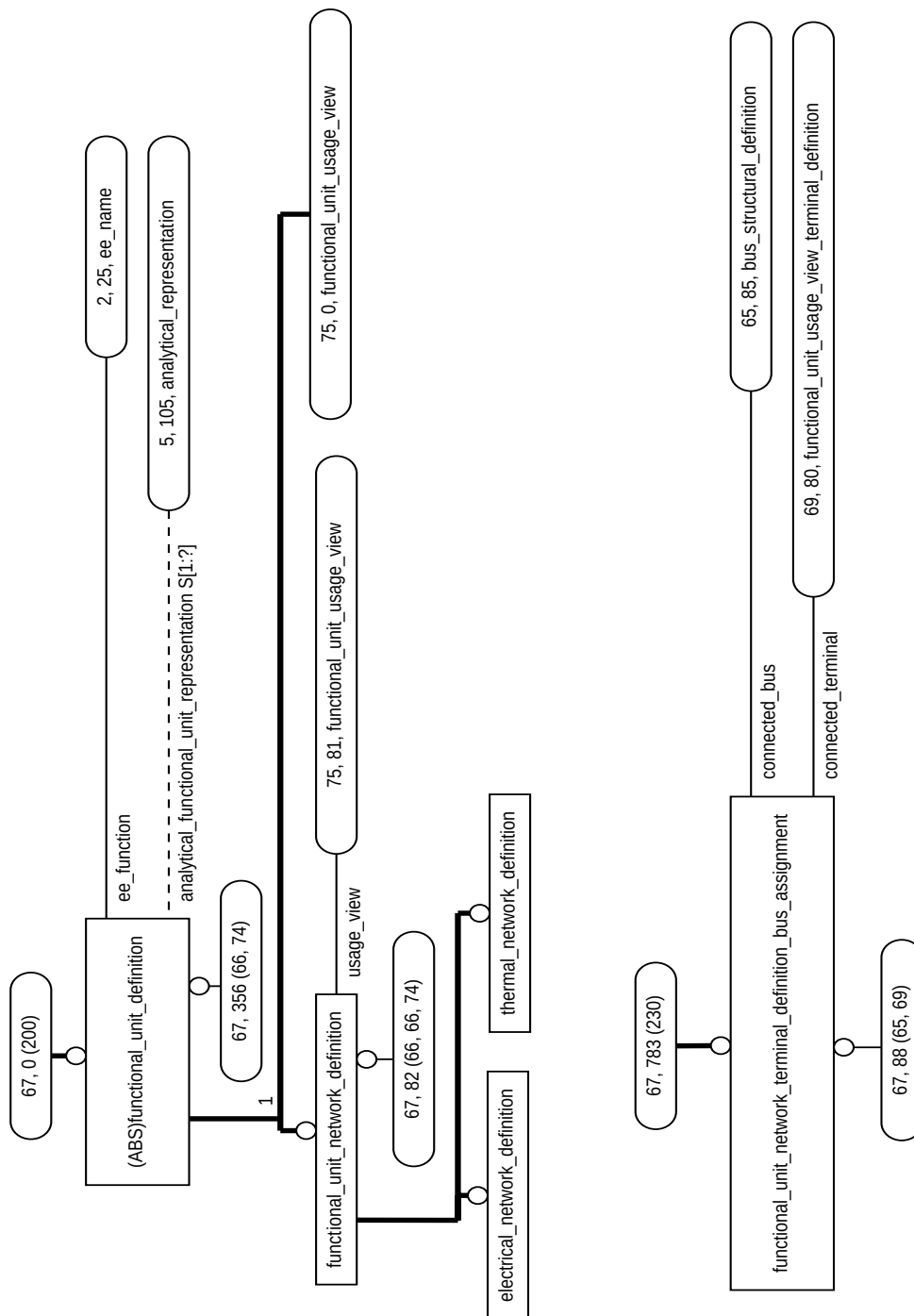


Figure H.67 – ARM EXPRESS-G Graphic 67 of 243

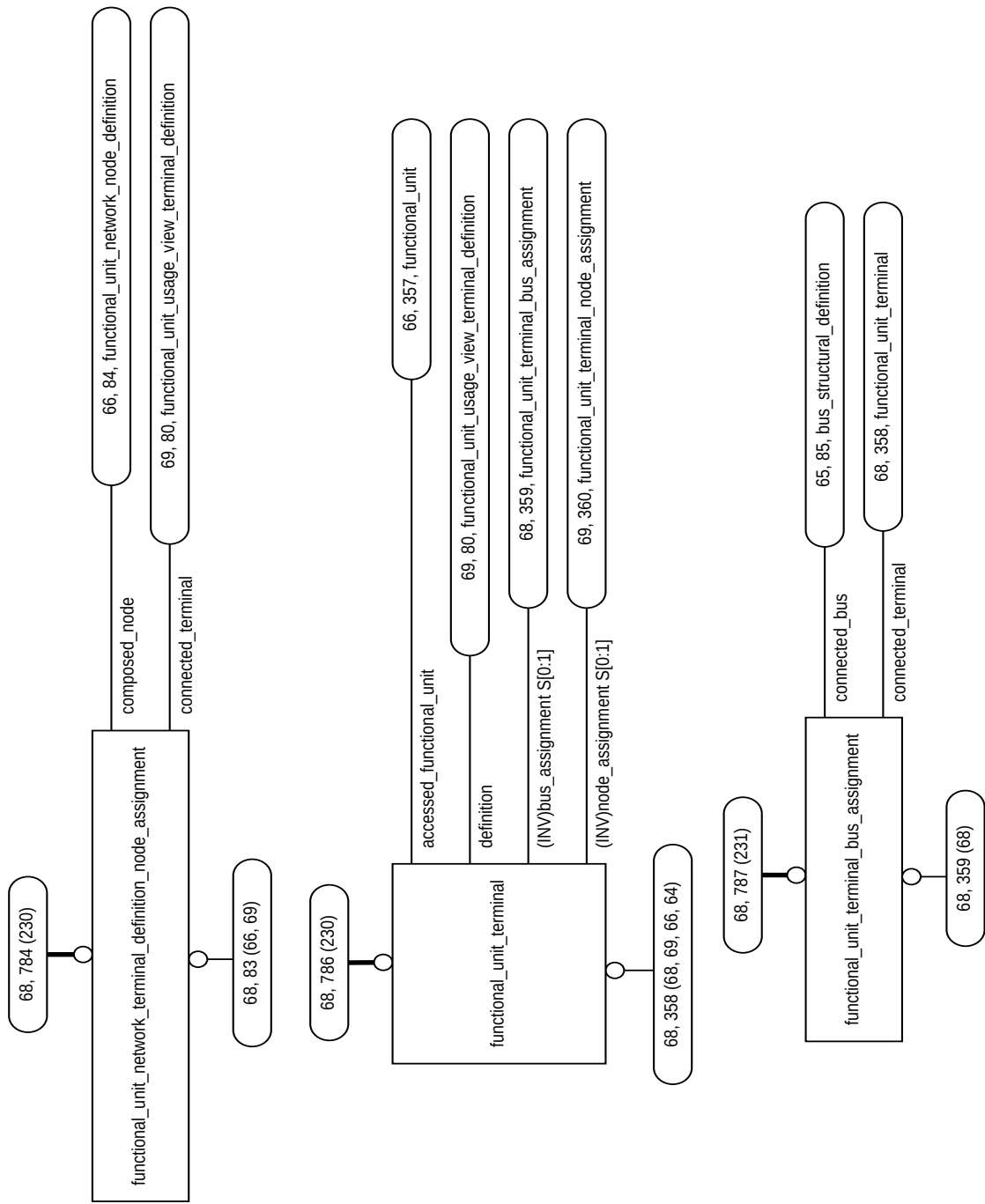


Figure H.68 – ARM EXPRESS-G Graphic 68 of 243

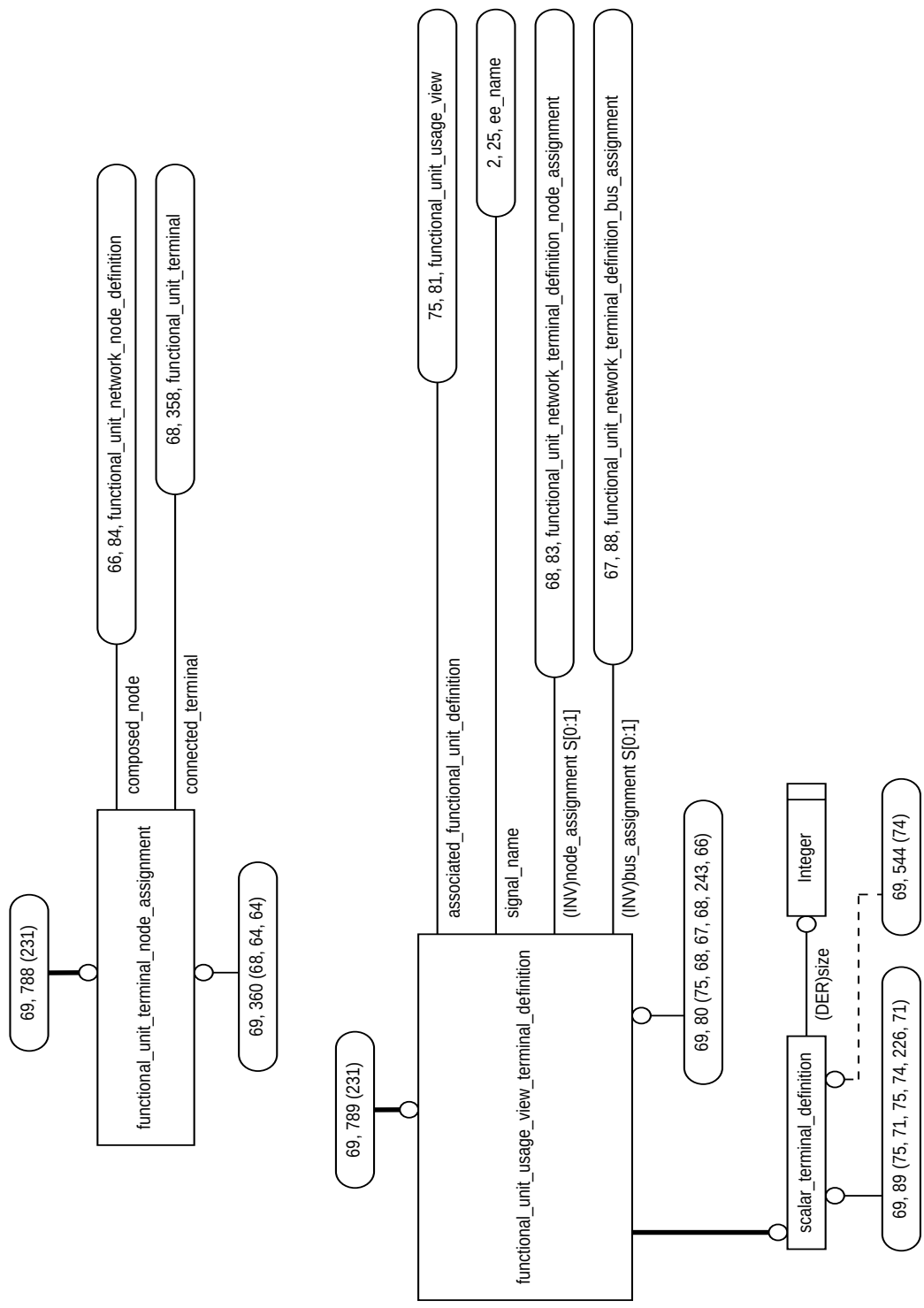


Figure H.69 – ARM EXPRESS-G Graphic 69 of 243

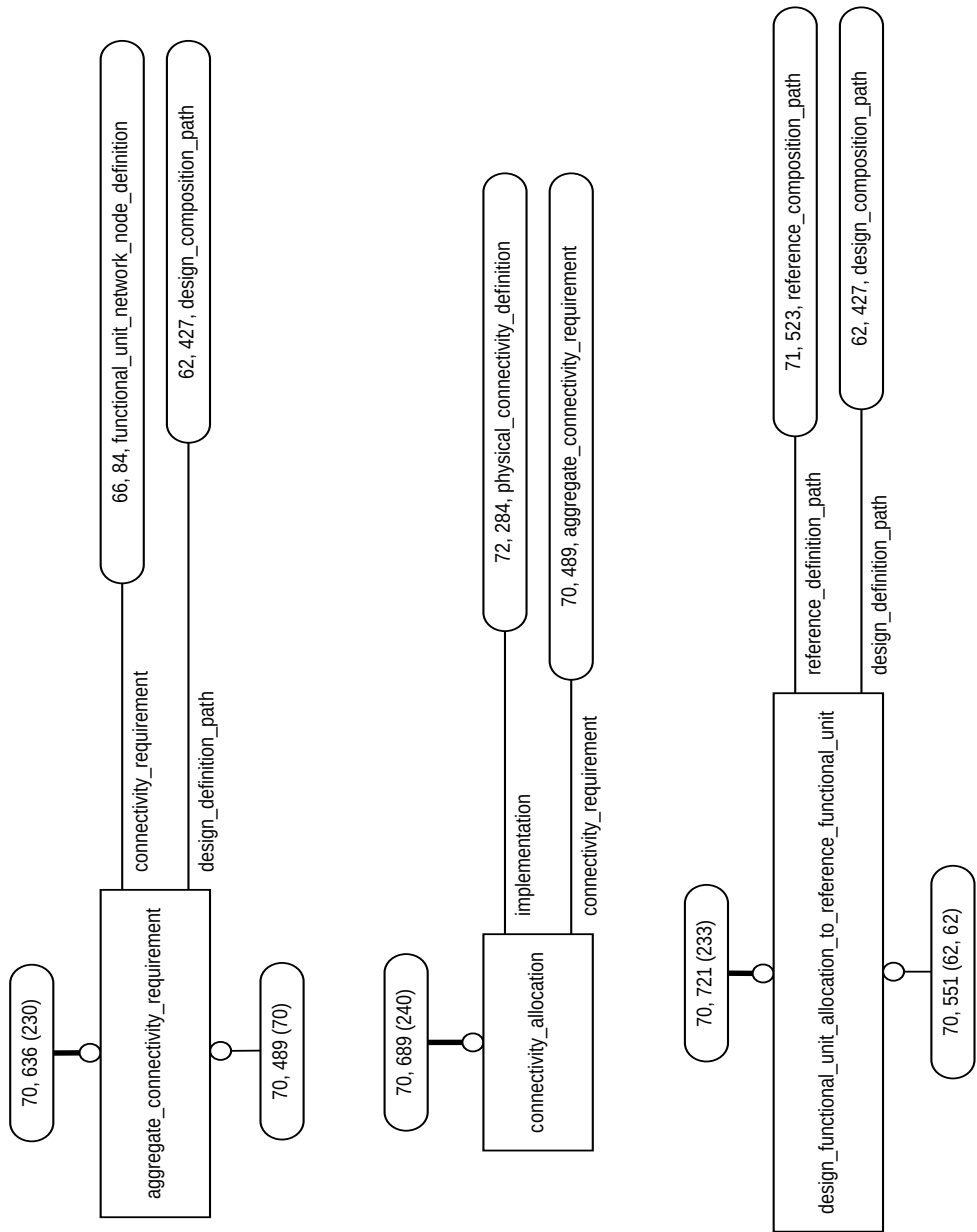


Figure H.70 – ARM EXPRESS-G Graphic 70 of 243

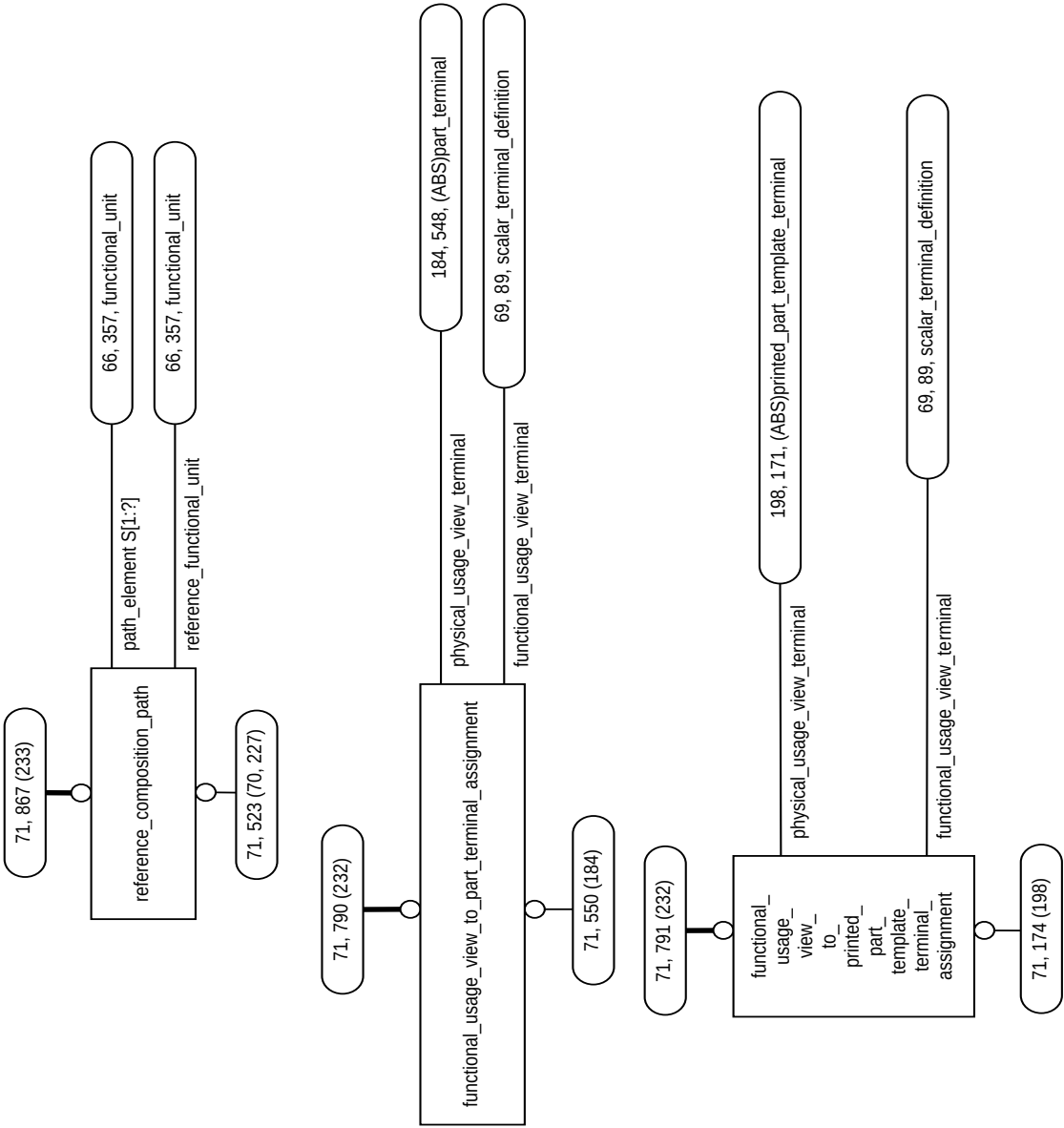


Figure H.71 – ARM EXPRESS-G Graphic 71 of 243

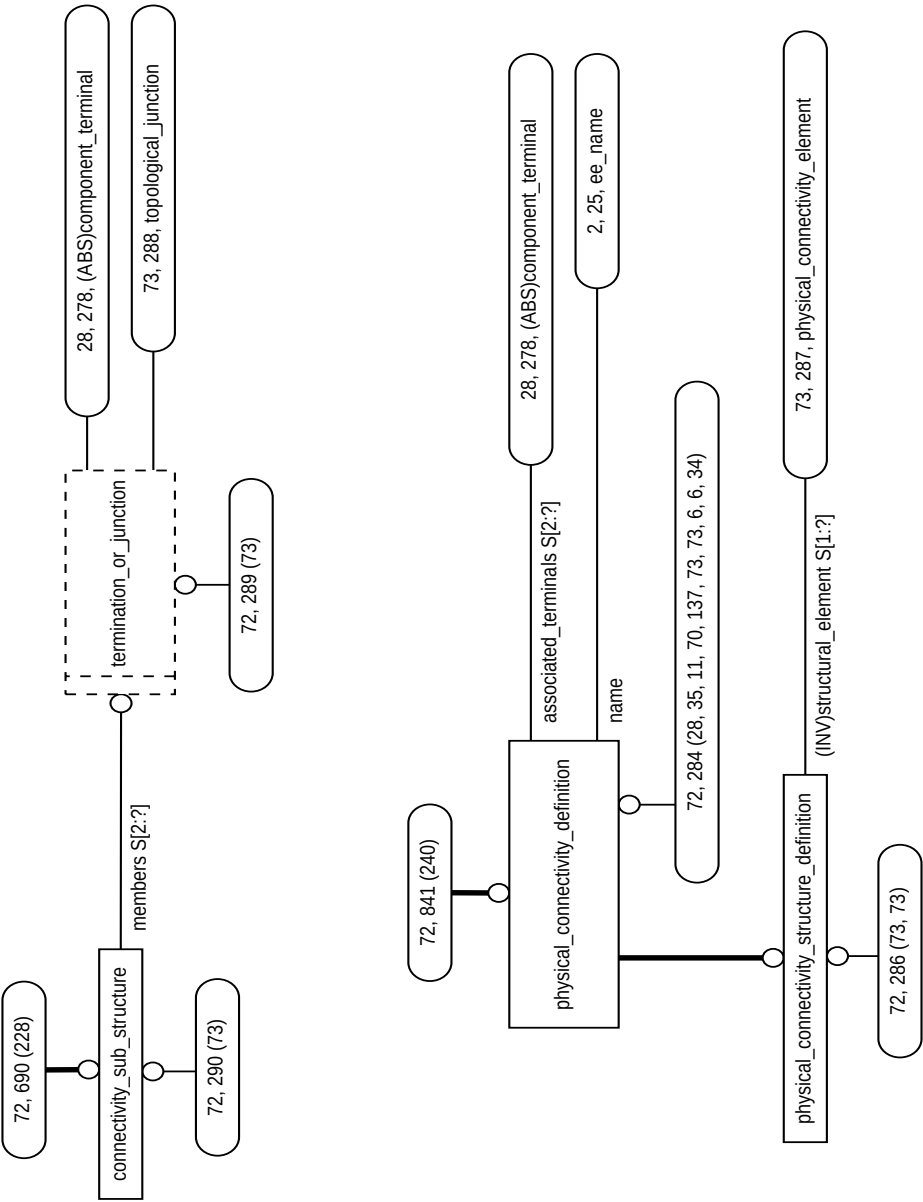


Figure H.72 – ARM EXPRESS-G Graphic 72 of 243

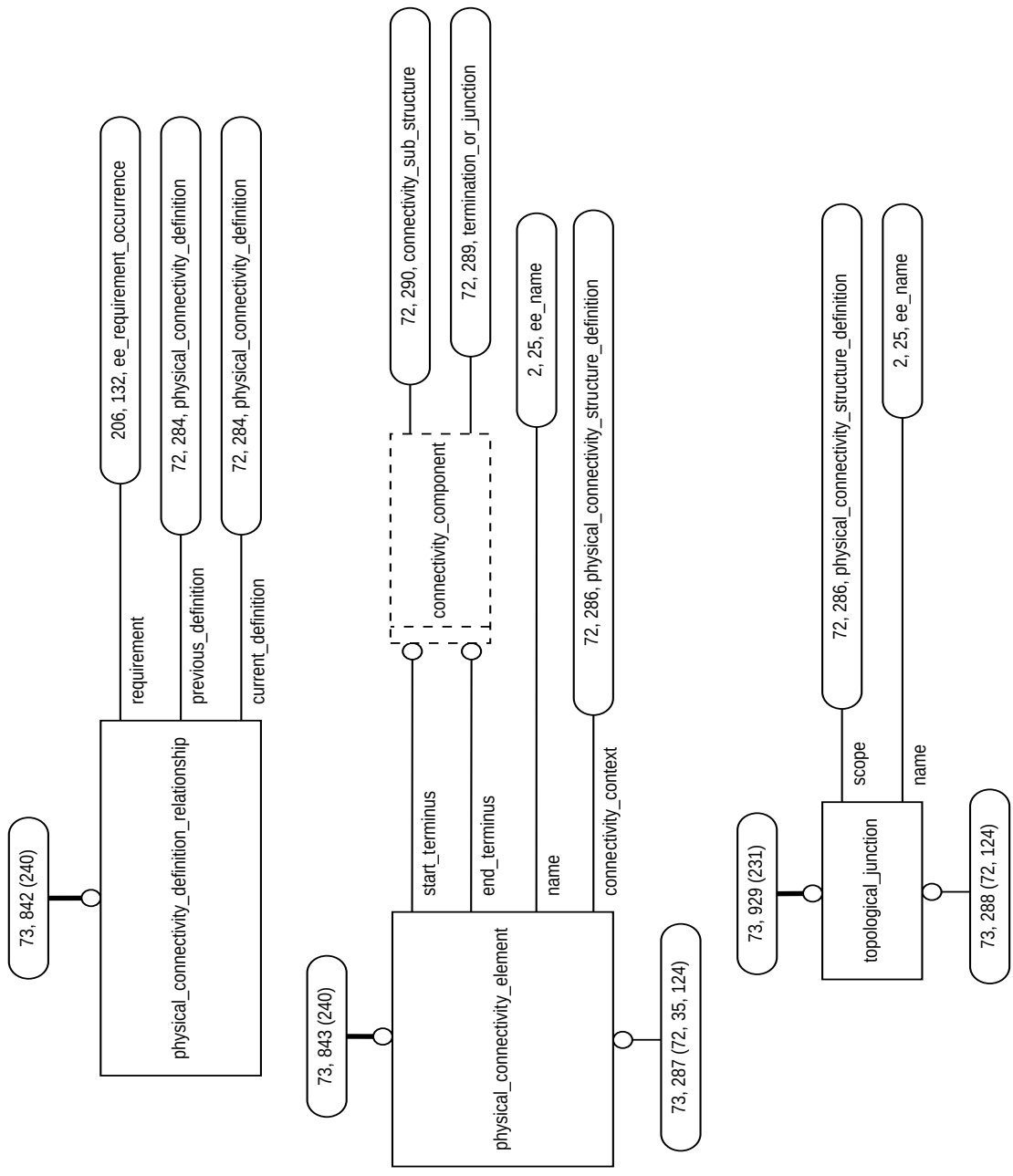


Figure H.73 – ARM EXPRESS-G Graphic 73 of 243

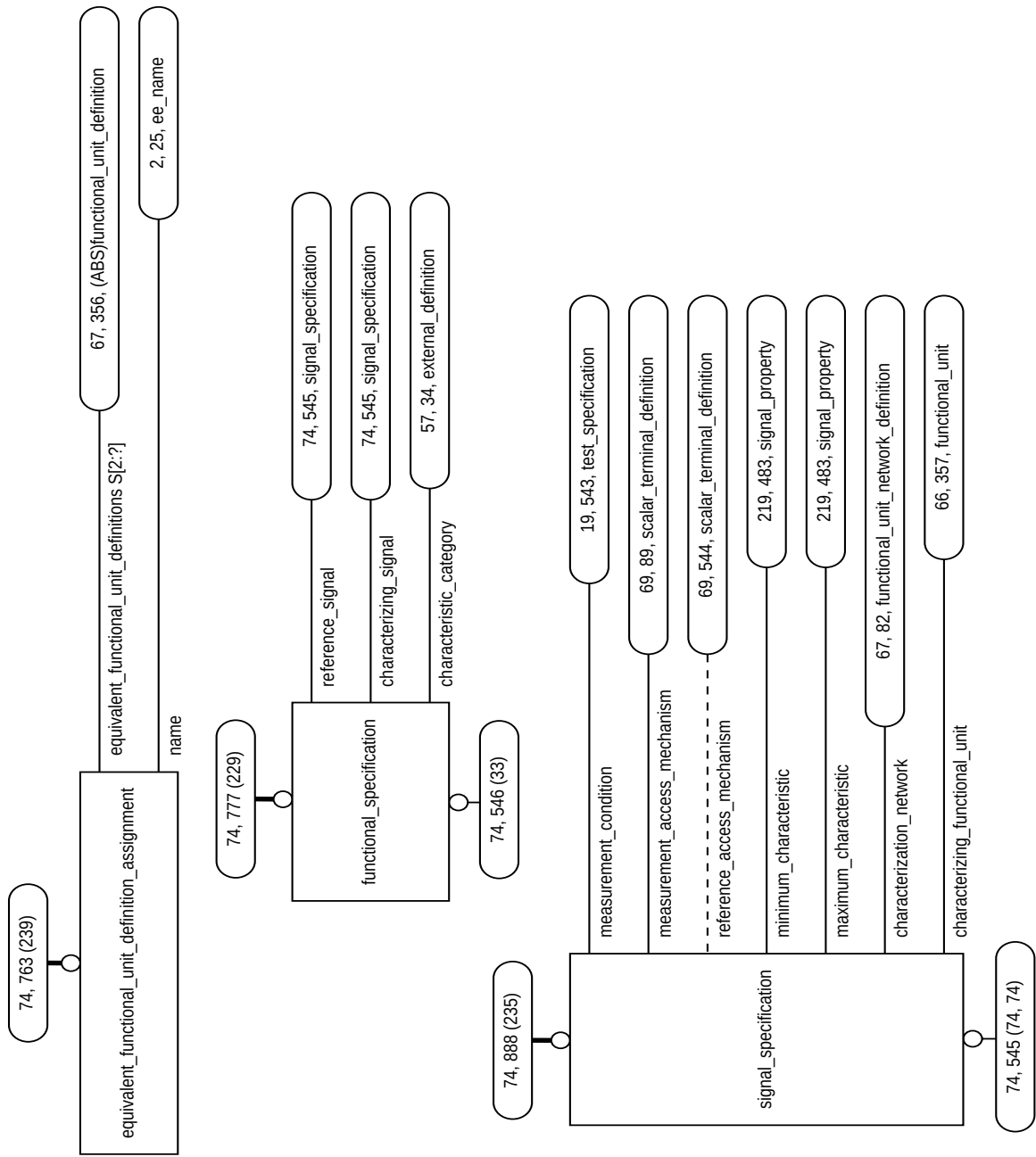


Figure H.74 – ARM EXPRESS-G Graphic 74 of 243

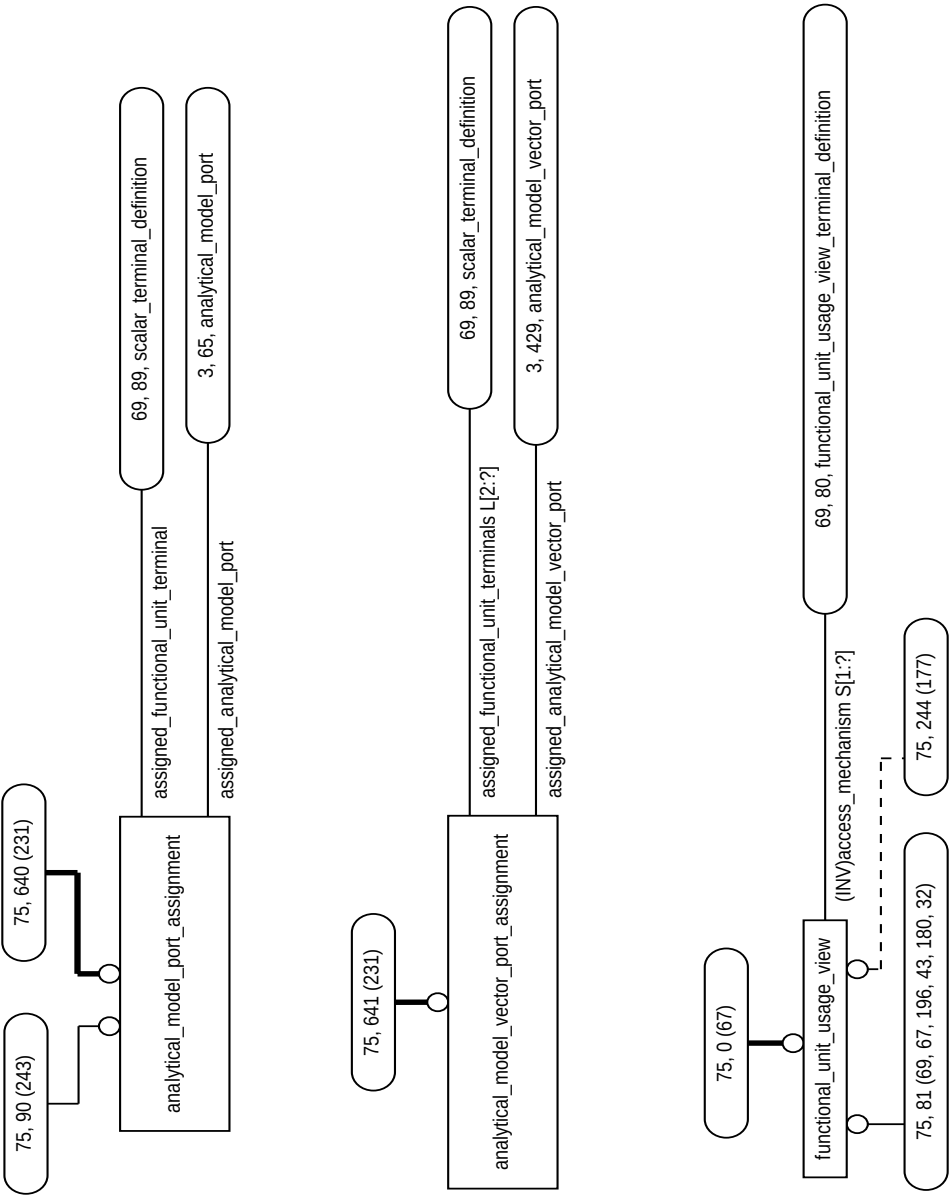


Figure H.75 – ARM EXPRESS-G Graphic 75 of 243

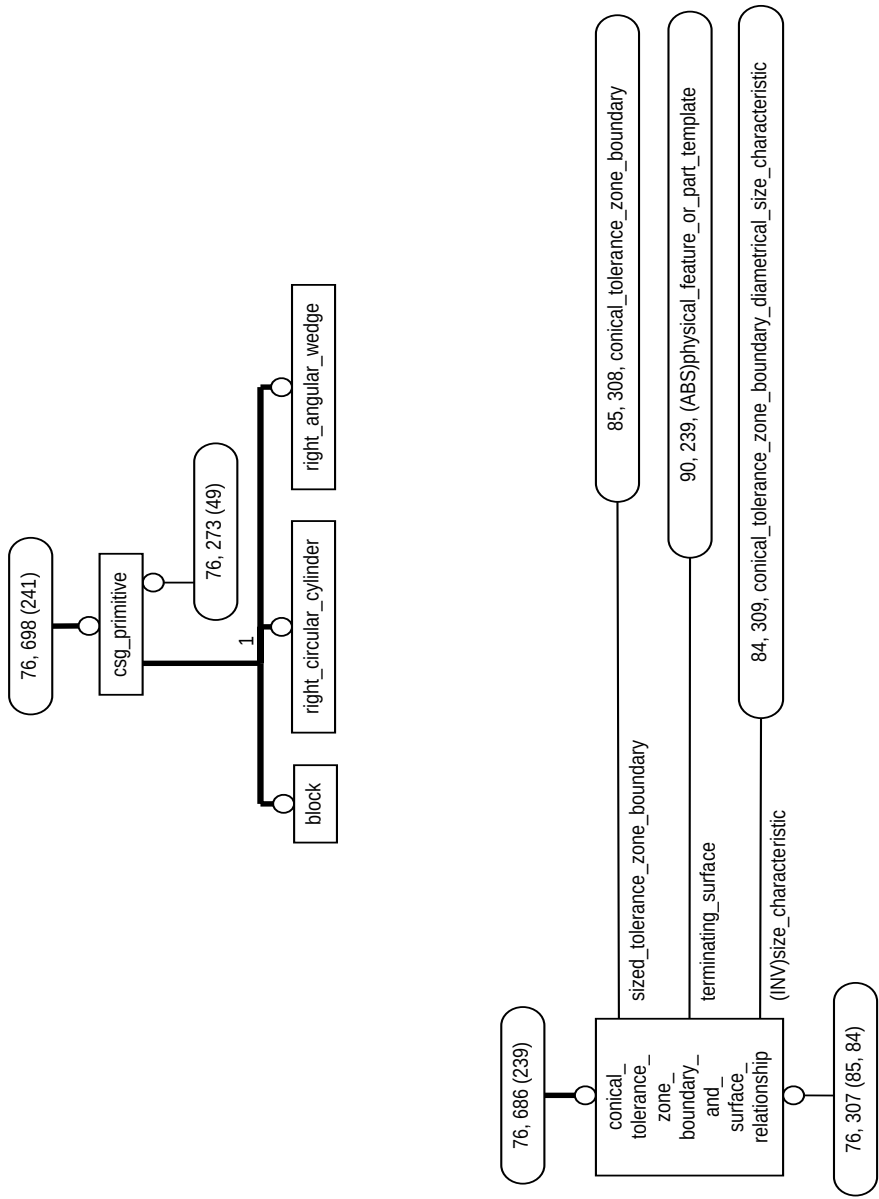


Figure H.76 – ARM EXPRESS-G Graphic 76 of 243

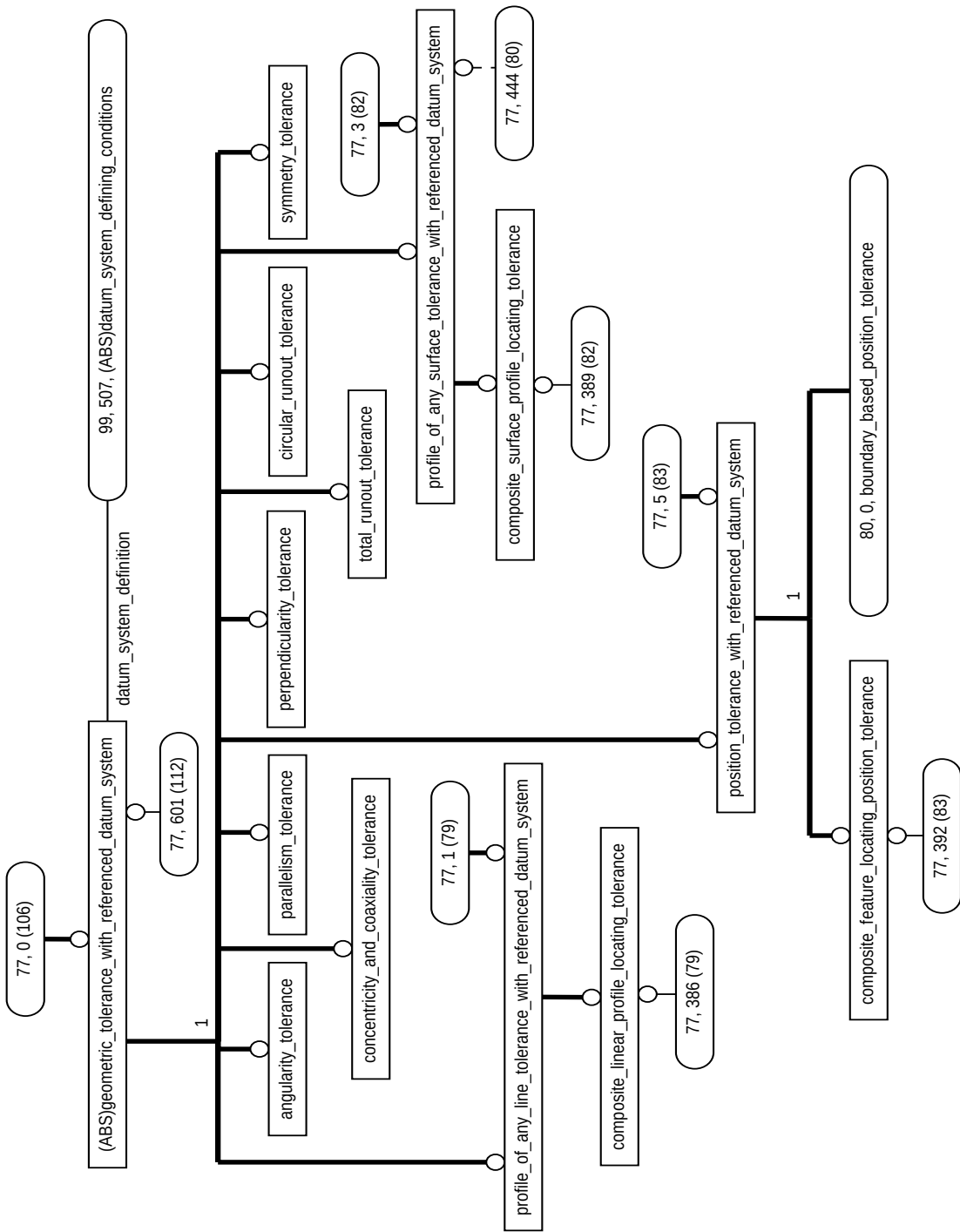


Figure H.77 – ARM EXPRESS-G Graphic 77 of 243

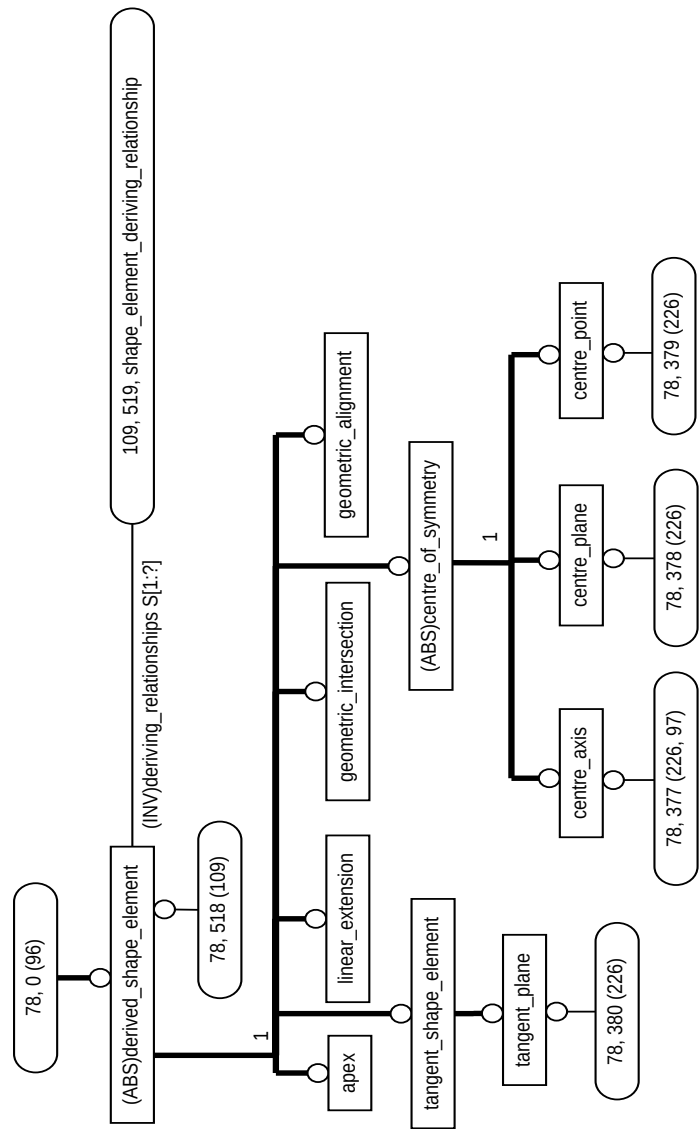


Figure H.78 – ARM EXPRESS-G Graphic 78 of 243

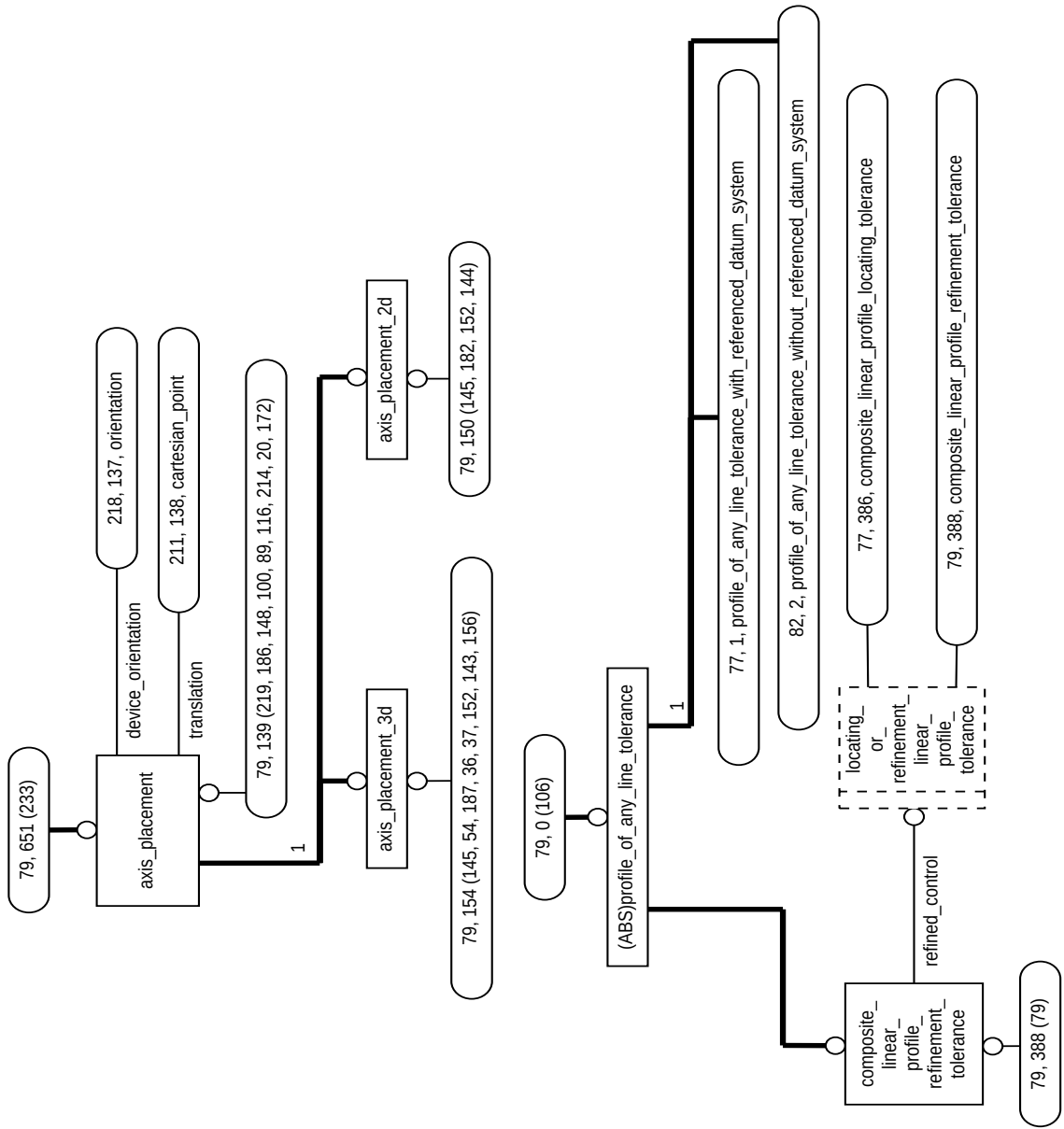


Figure H.79 – ARM EXPRESS-G Graphic 79 of 243

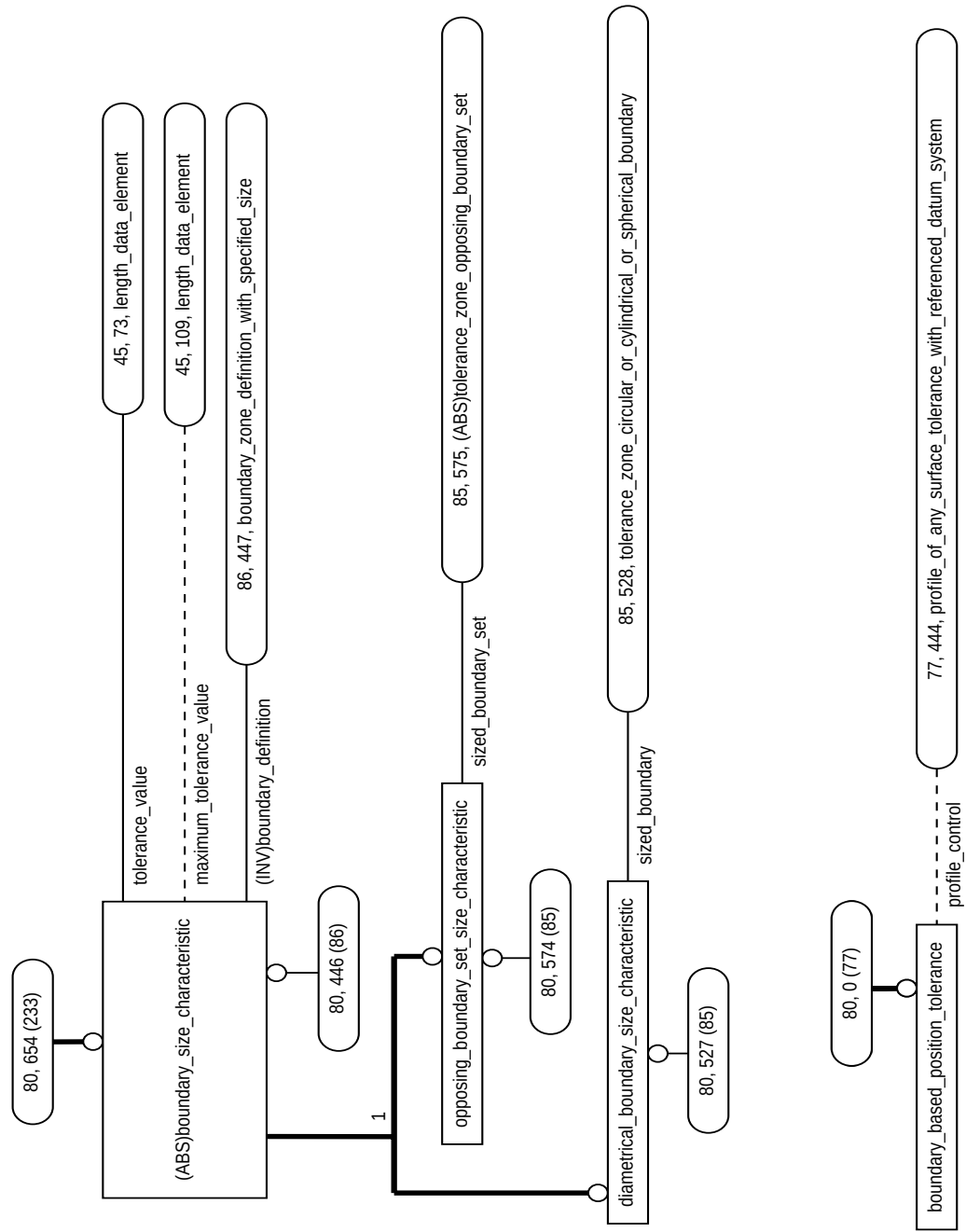


Figure H.80 – ARM EXPRESS-G Graphic 80 of 243

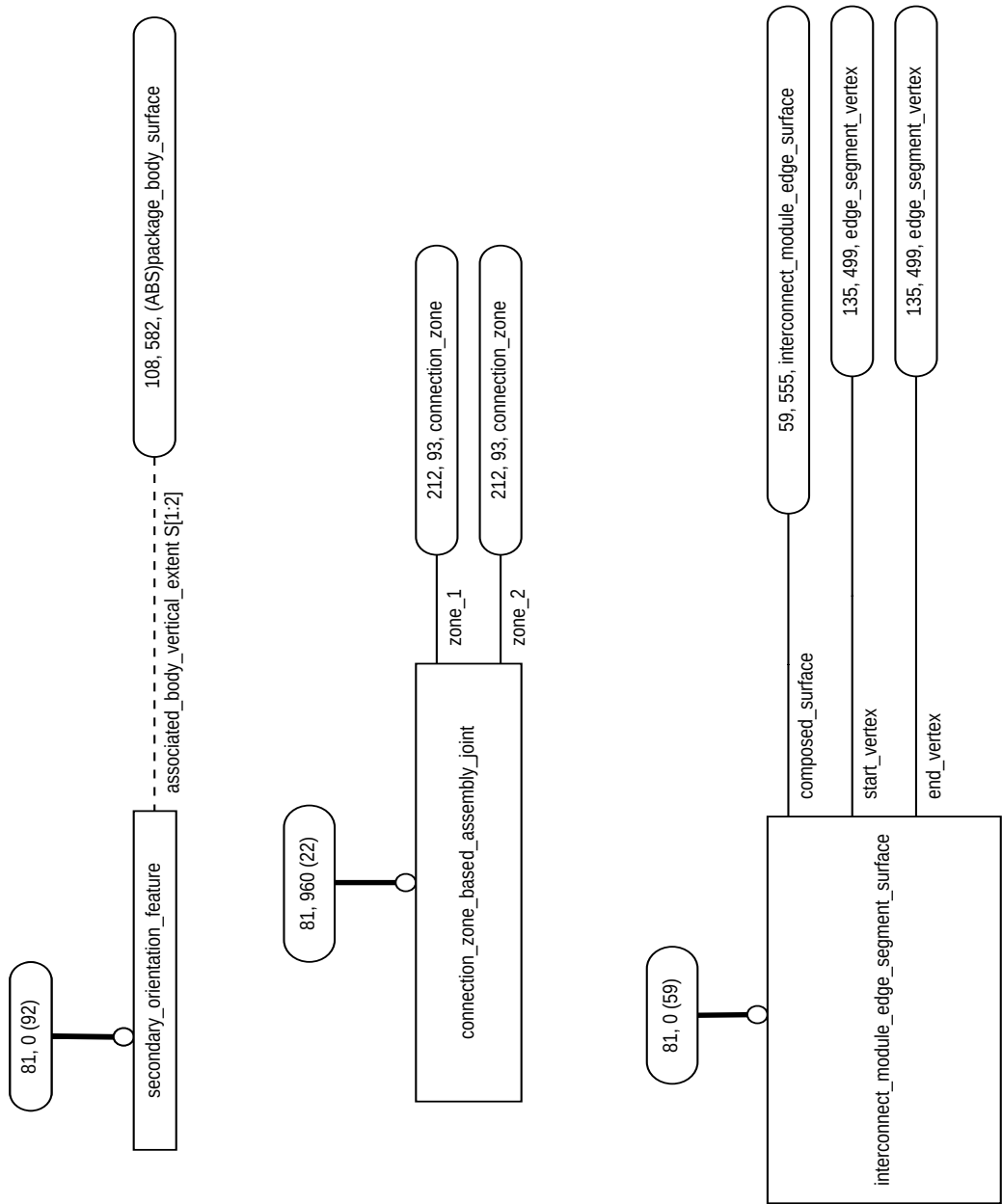


Figure H.81 – ARM EXPRESS-G Graphic 81 of 243

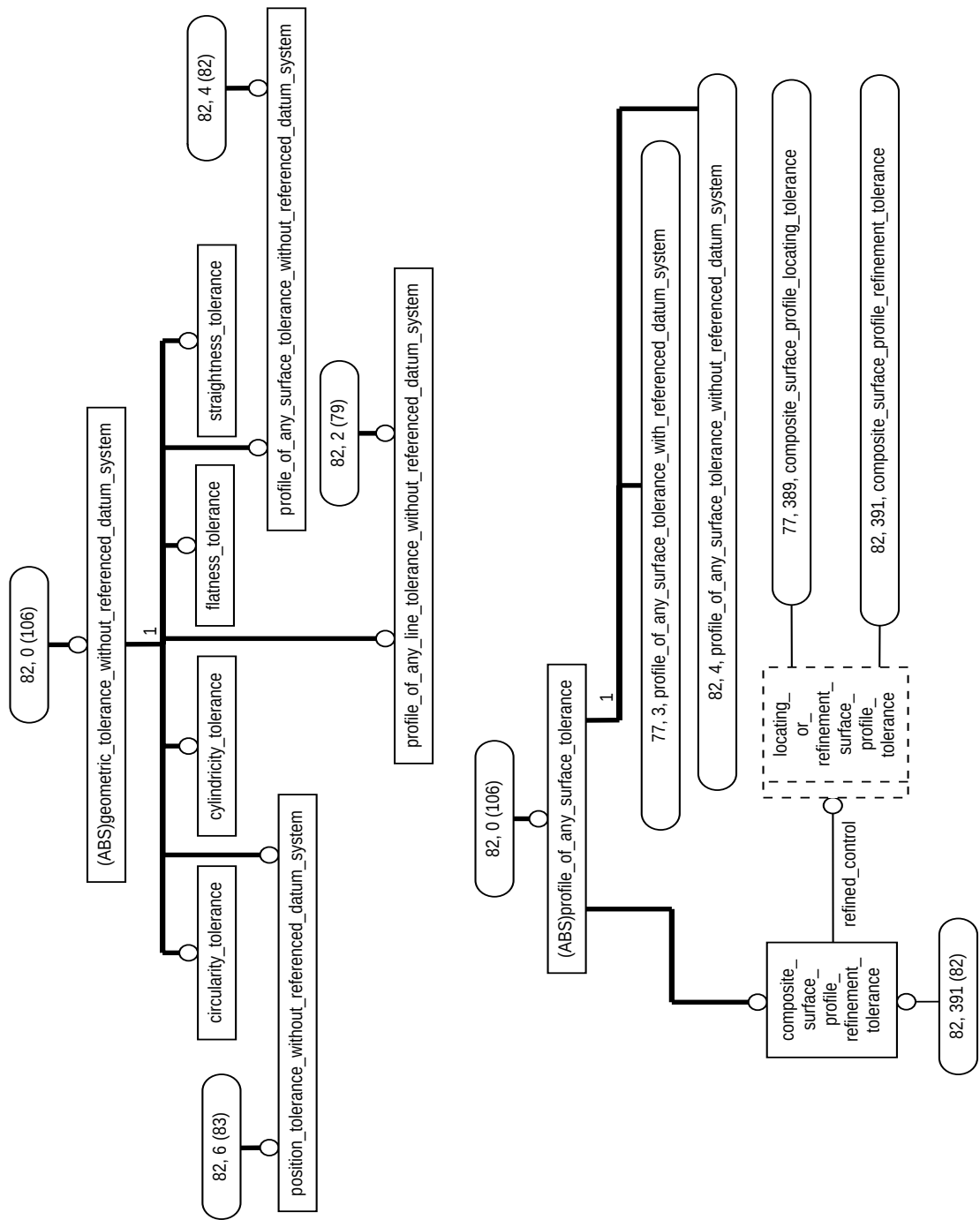


Figure H.82 – ARM EXPRESS-G Graphic 82 of 243

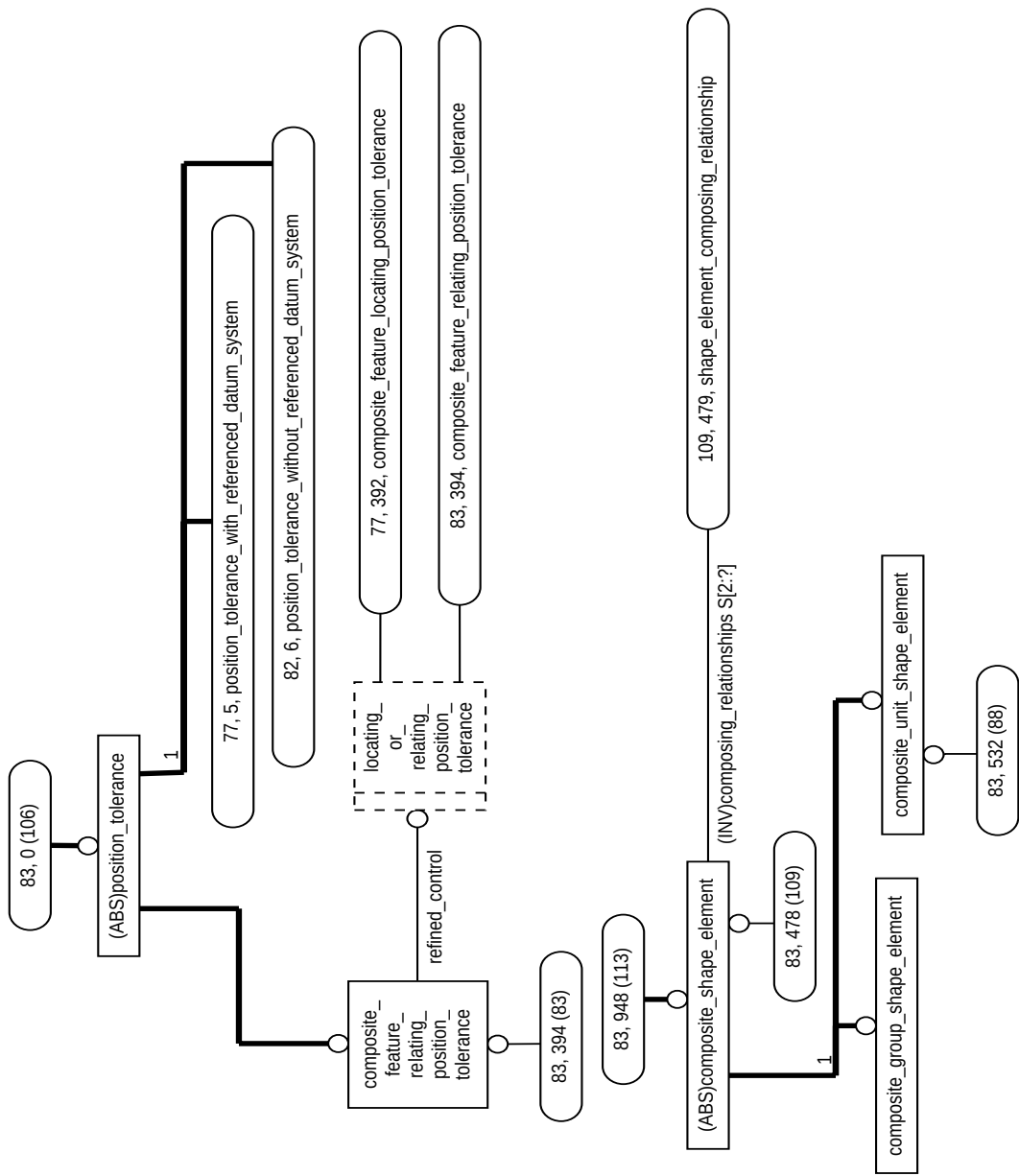


Figure H.83 – ARM EXPRESS-G Graphic 83 of 243

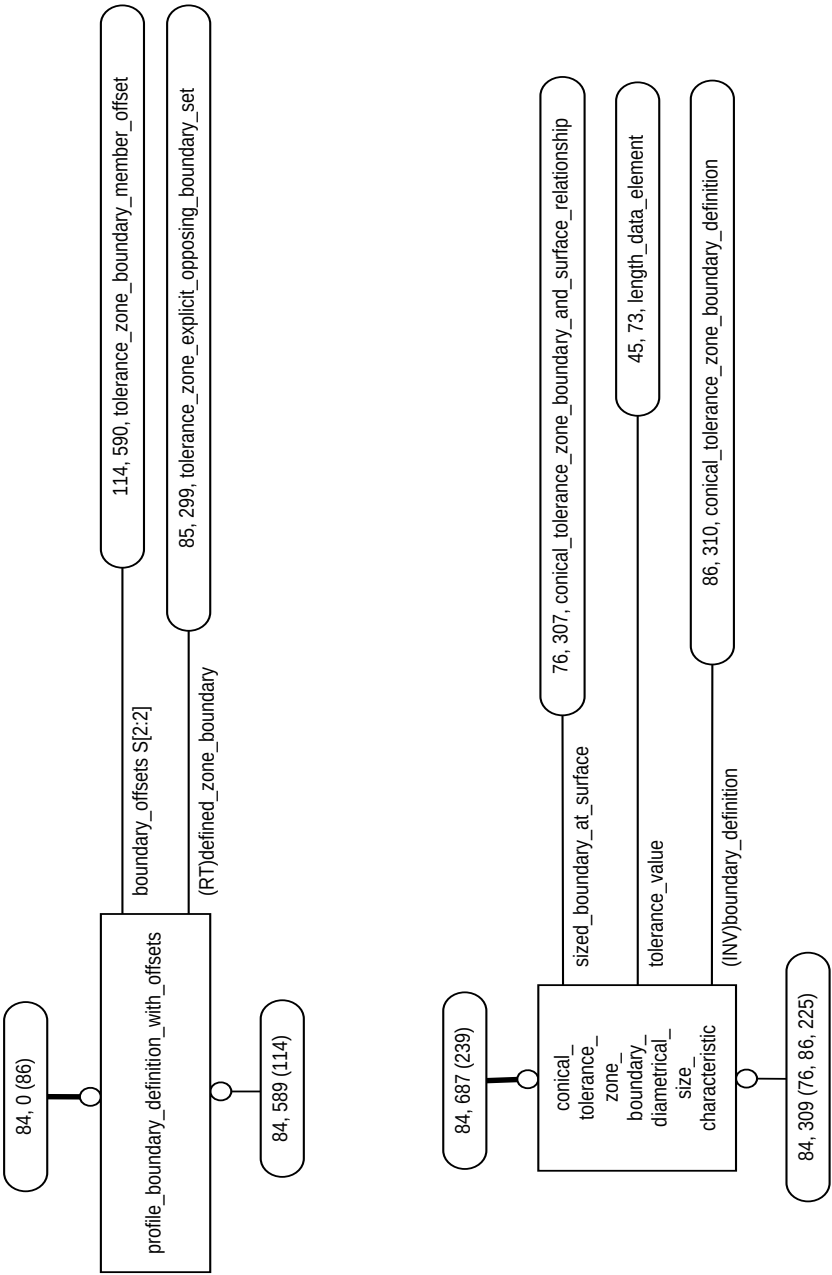


Figure H.84 – ARM EXPRESS-G Graphic 84 of 243

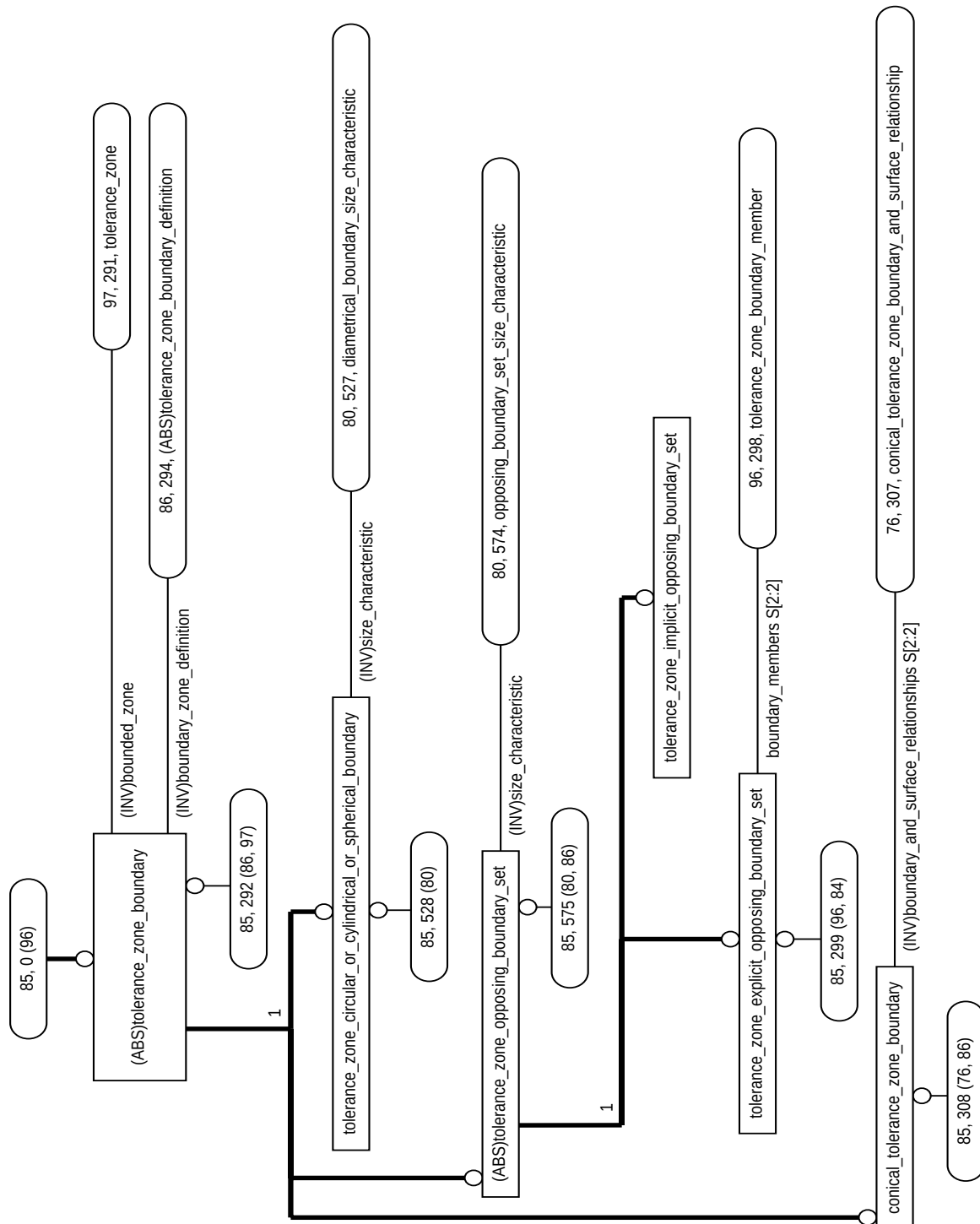


Figure H.85 – ARM EXPRESS-G Graphic 85 of 243

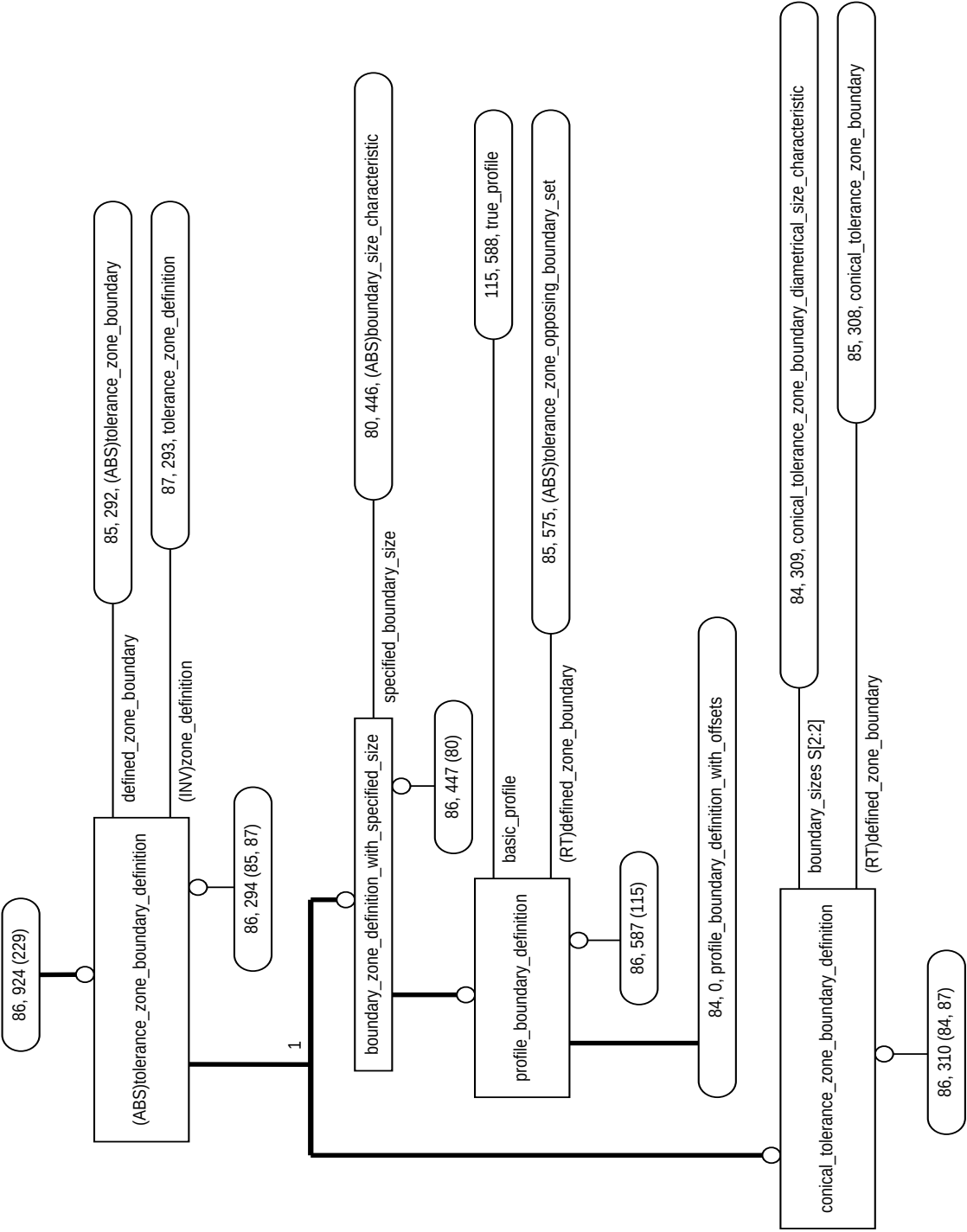


Figure H.86 – ARM EXPRESS-G Graphic 86 of 243

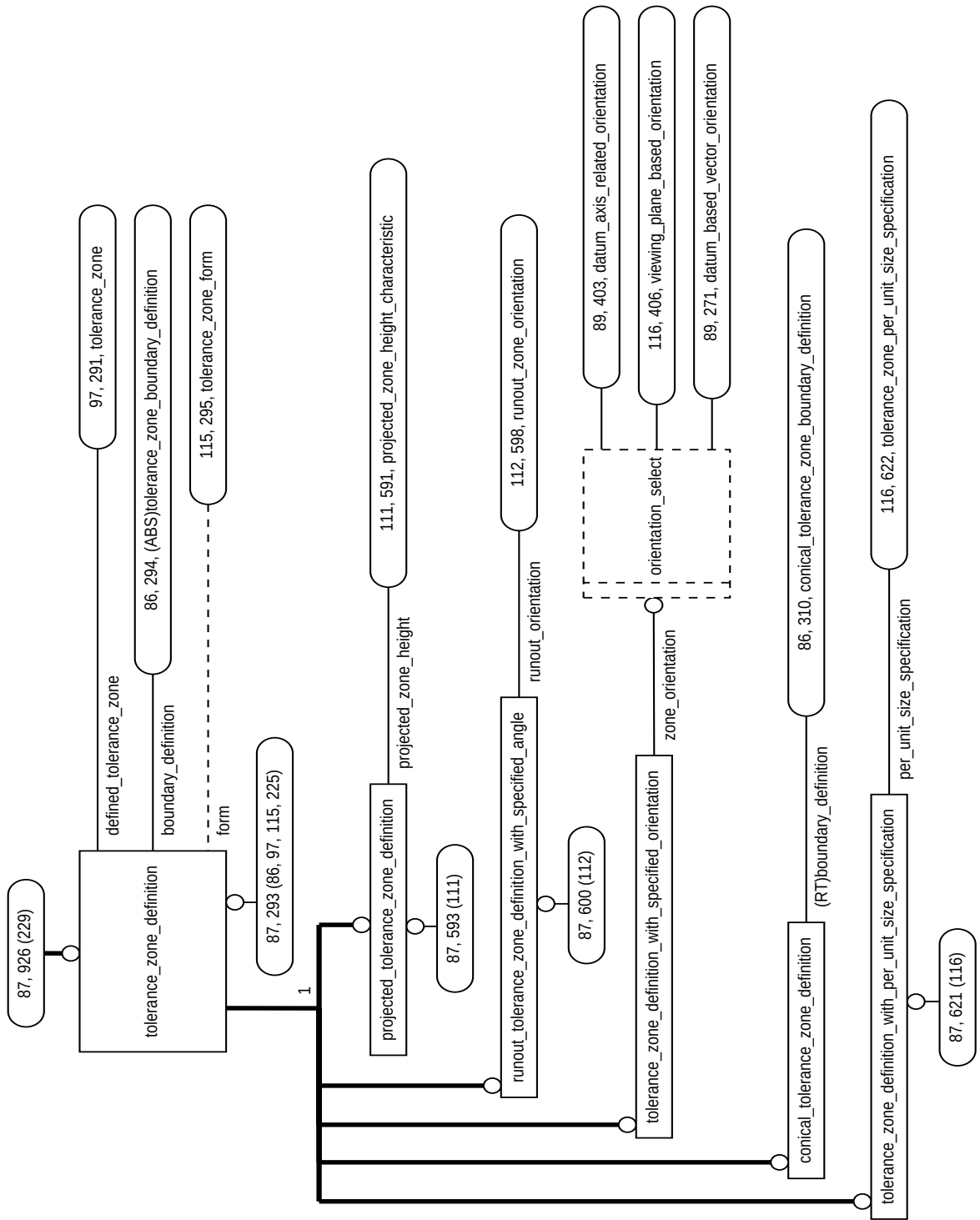


Figure H.87 – ARM EXPRESS-G Graphic 87 of 243

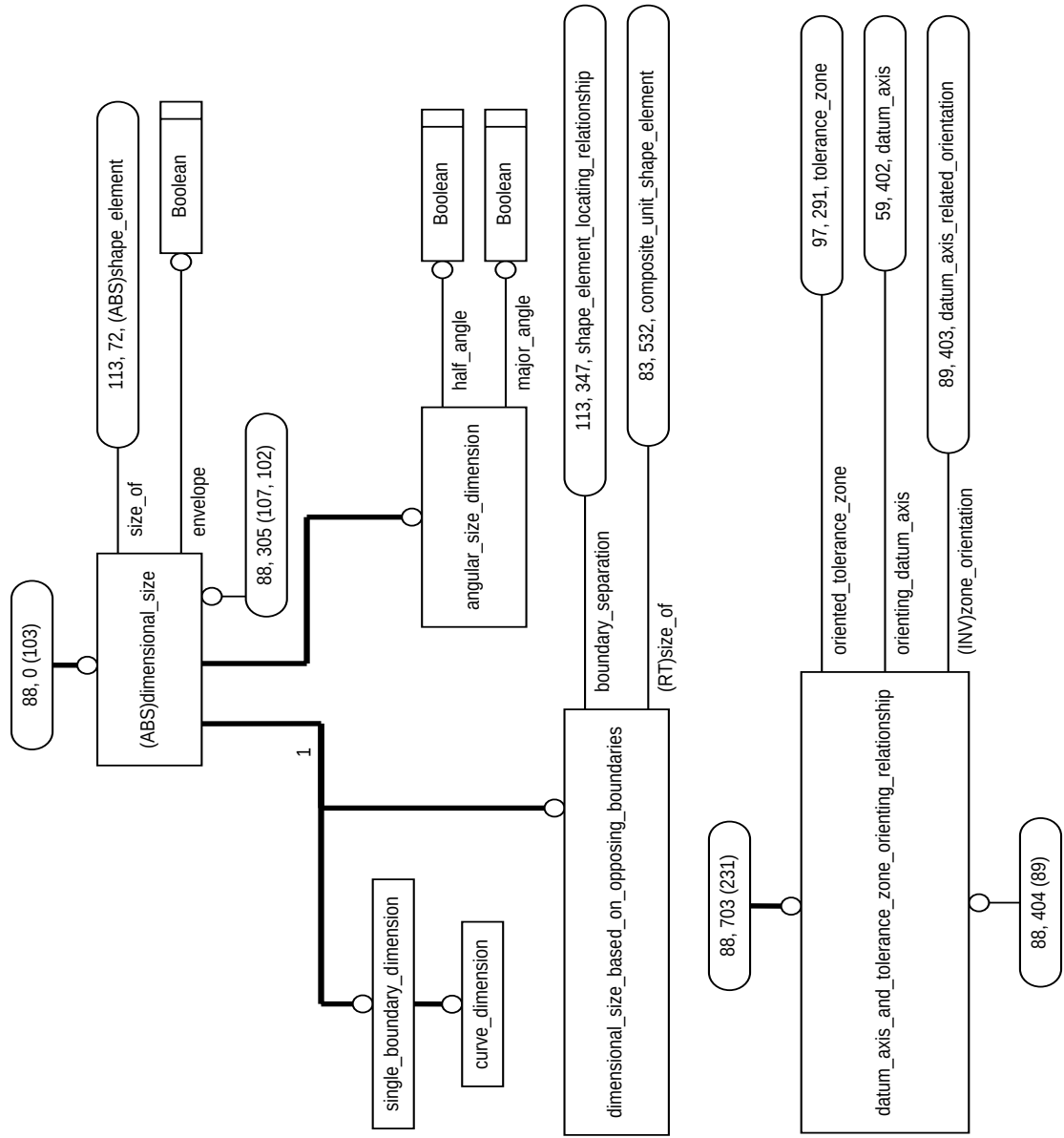


Figure H.88 – ARM EXPRESS-G Graphic 88 of 243

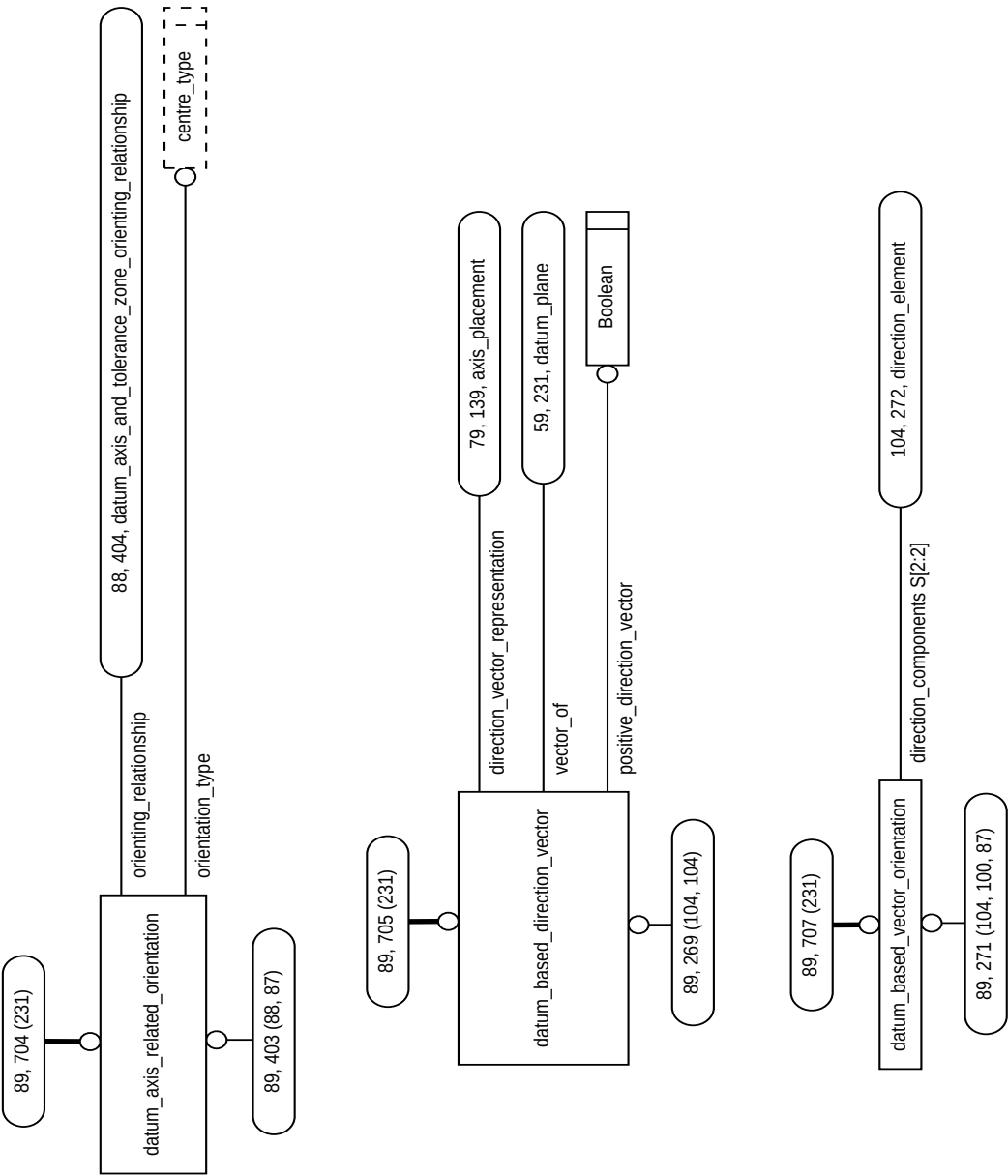


Figure H.89 – ARM EXPRESS-G Graphic 89 of 243

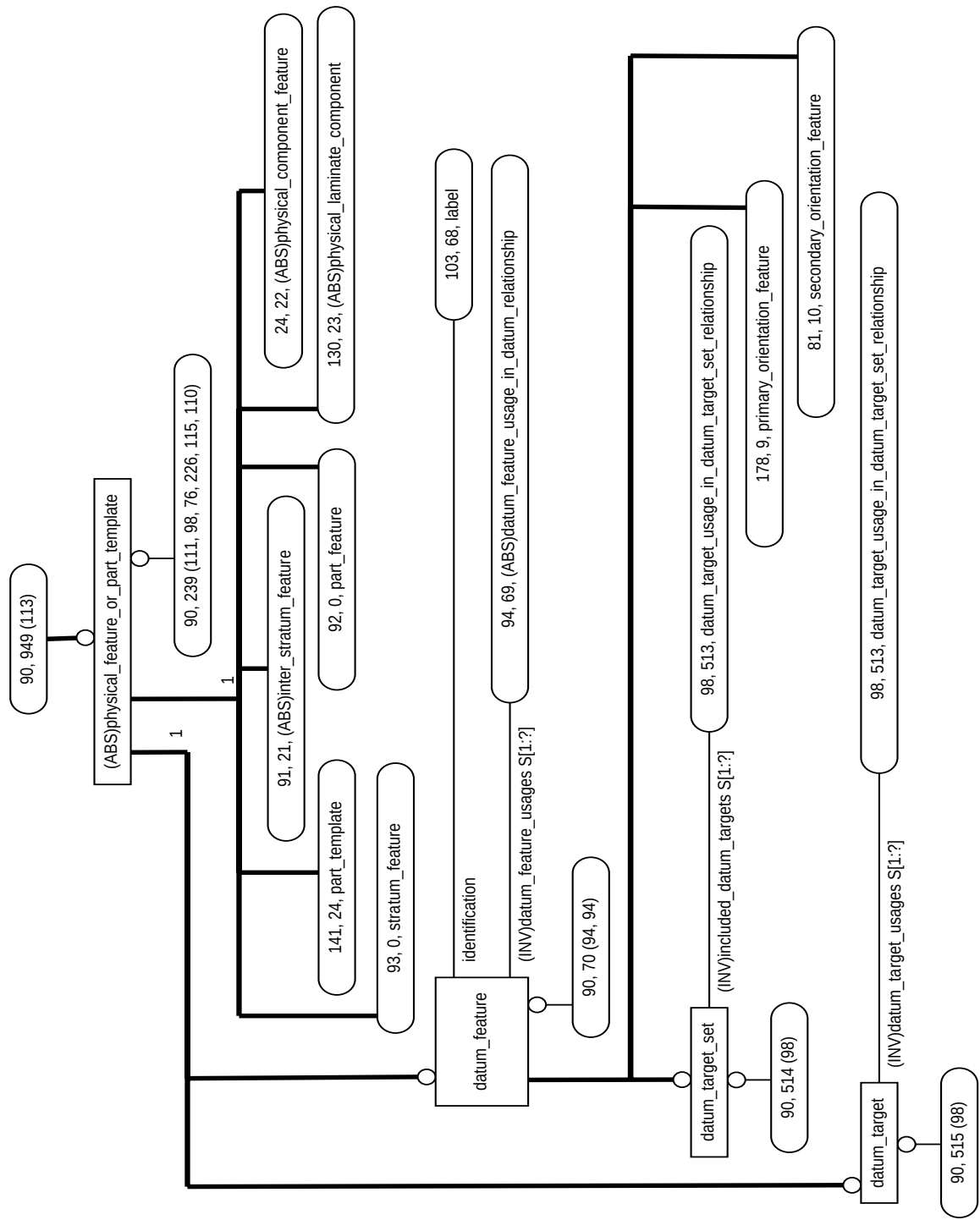


Figure H.90 – ARM EXPRESS-G Graphic 90 of 243

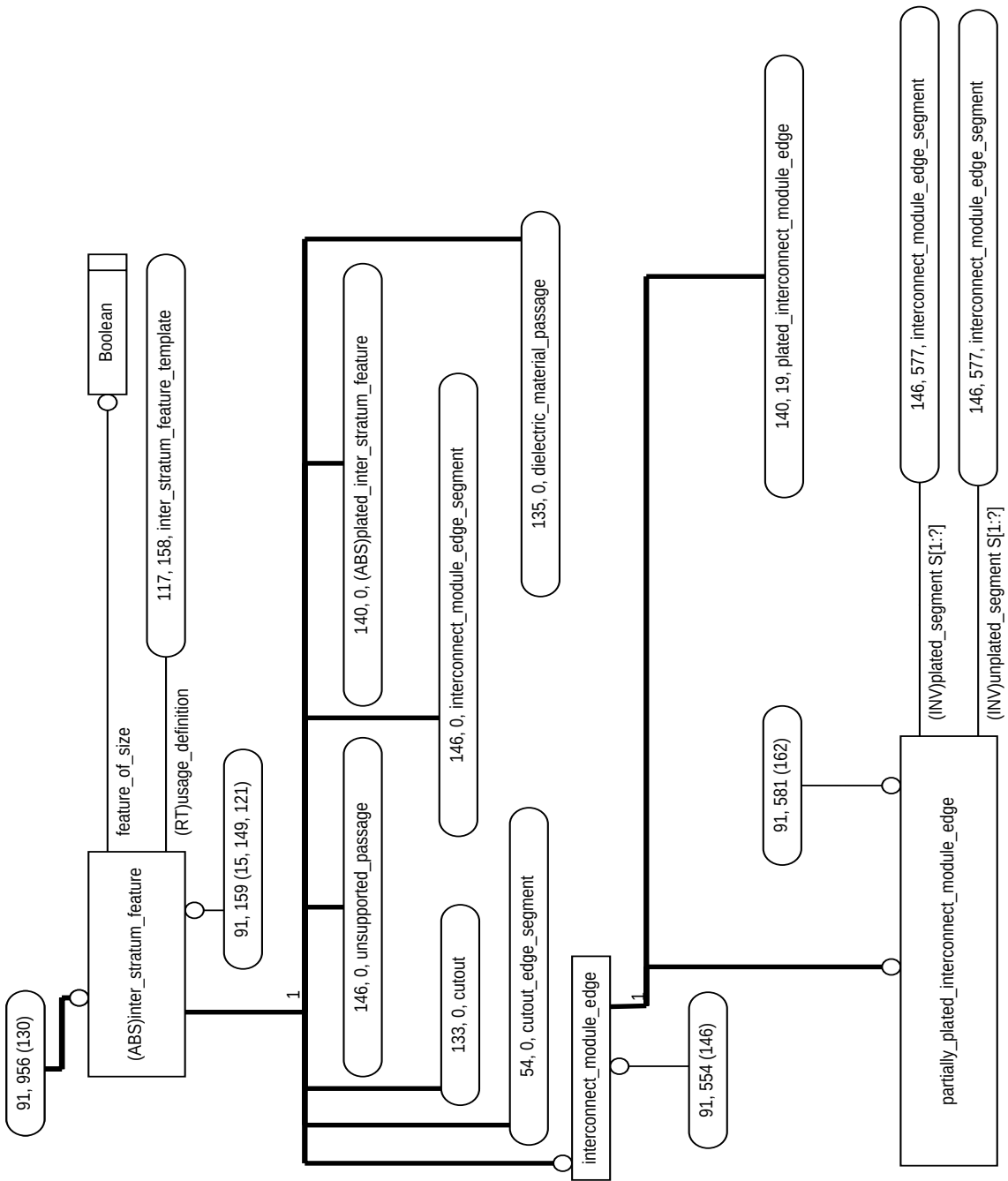


Figure H.91 – ARM EXPRESS-G Graphic 91 of 243

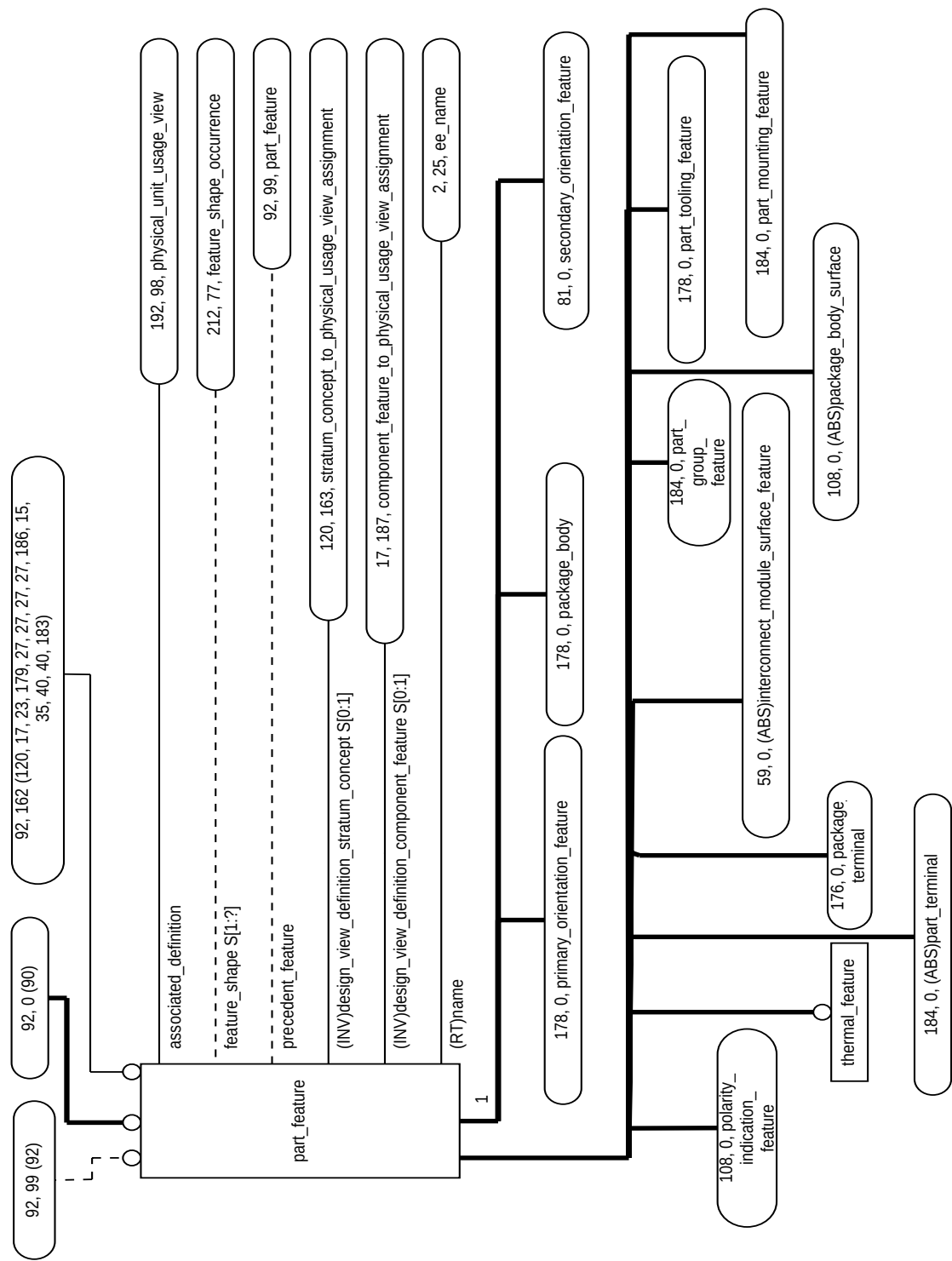


Figure H.92 – ARM EXPRESS-G Graphic 92 of 243

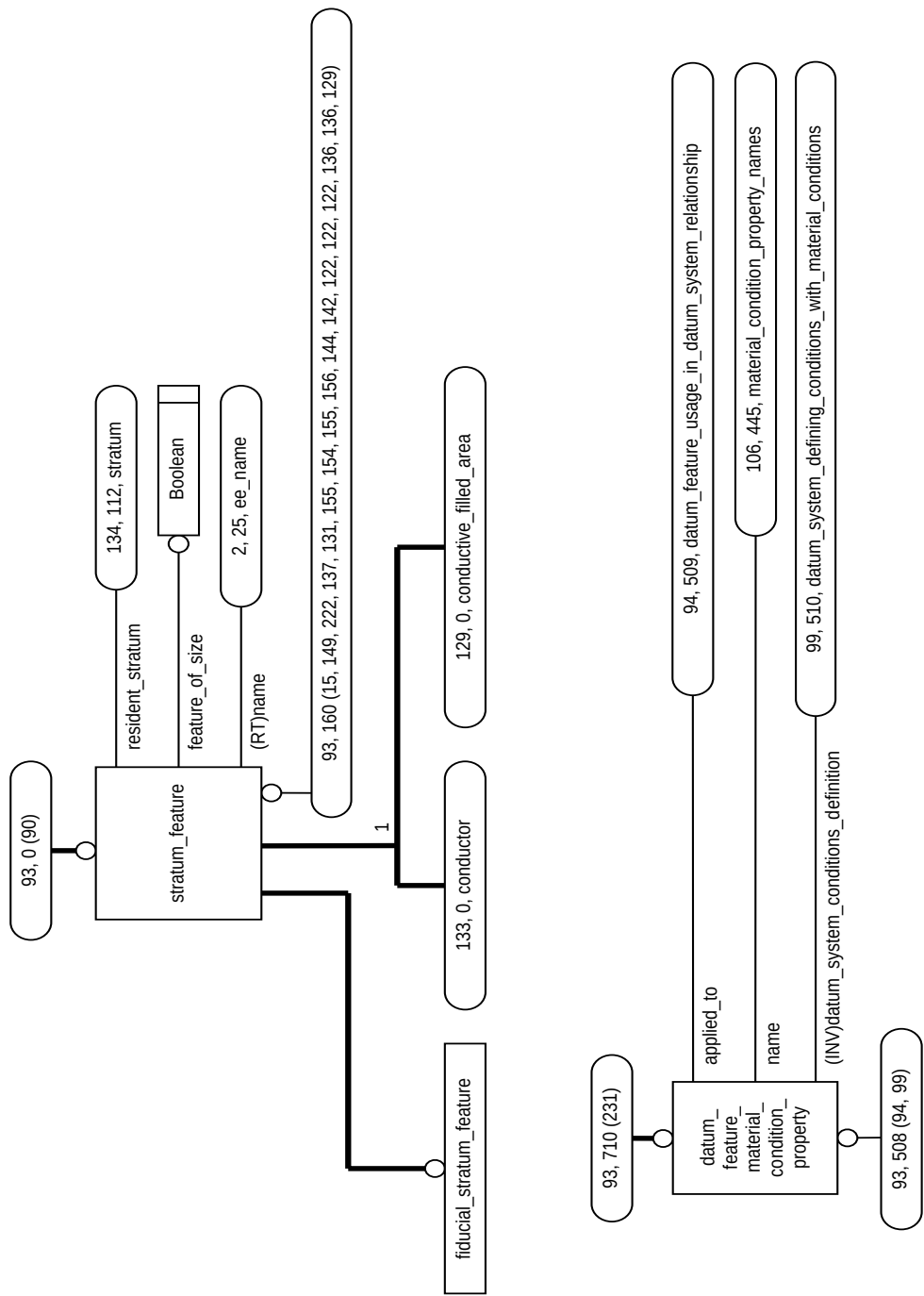


Figure H.93 – ARM EXPRESS-G Graphic 93 of 243

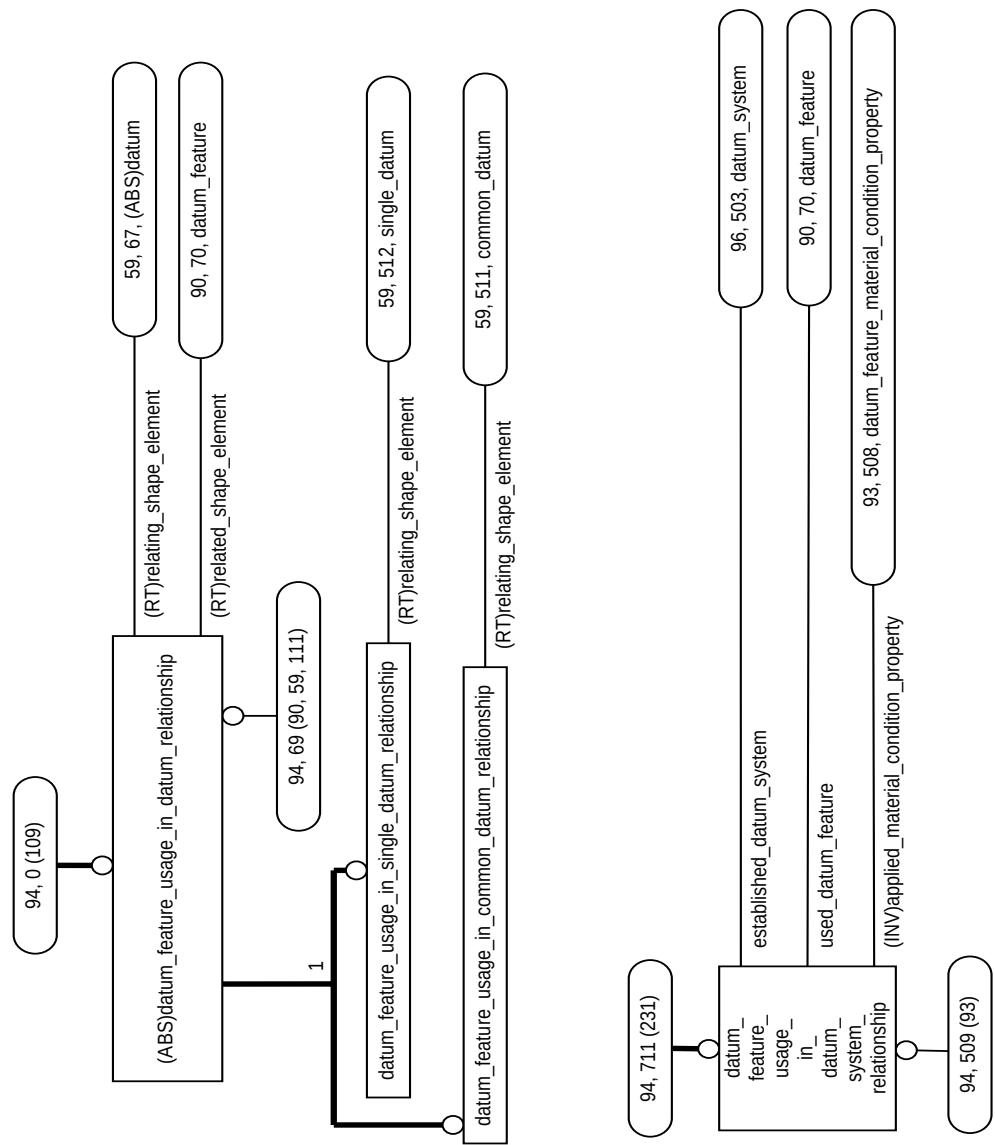


Figure H.94 – ARM EXPRESS-G Graphic 94 of 243

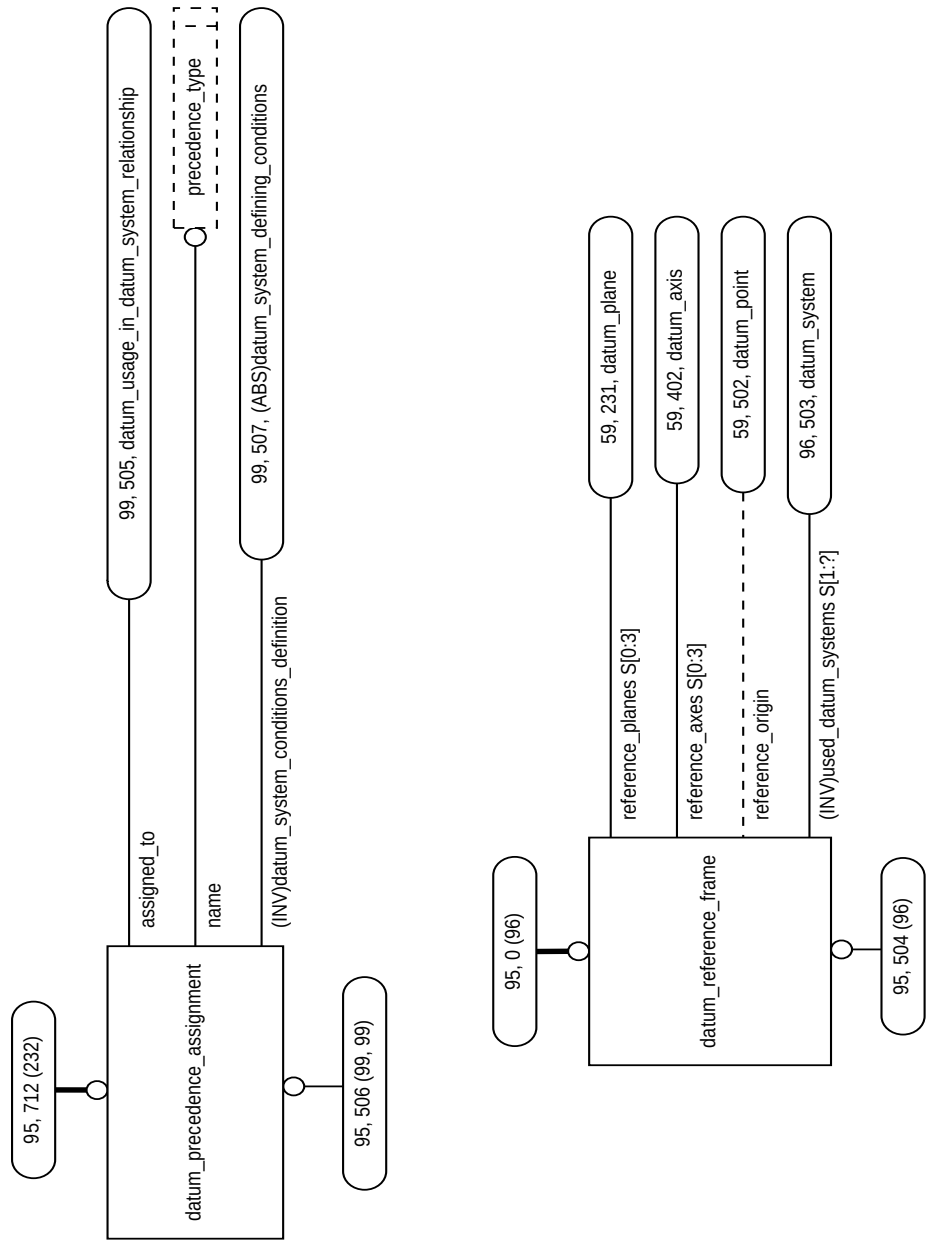


Figure H.95 – ARM EXPRESS-G Graphic 95 of 243



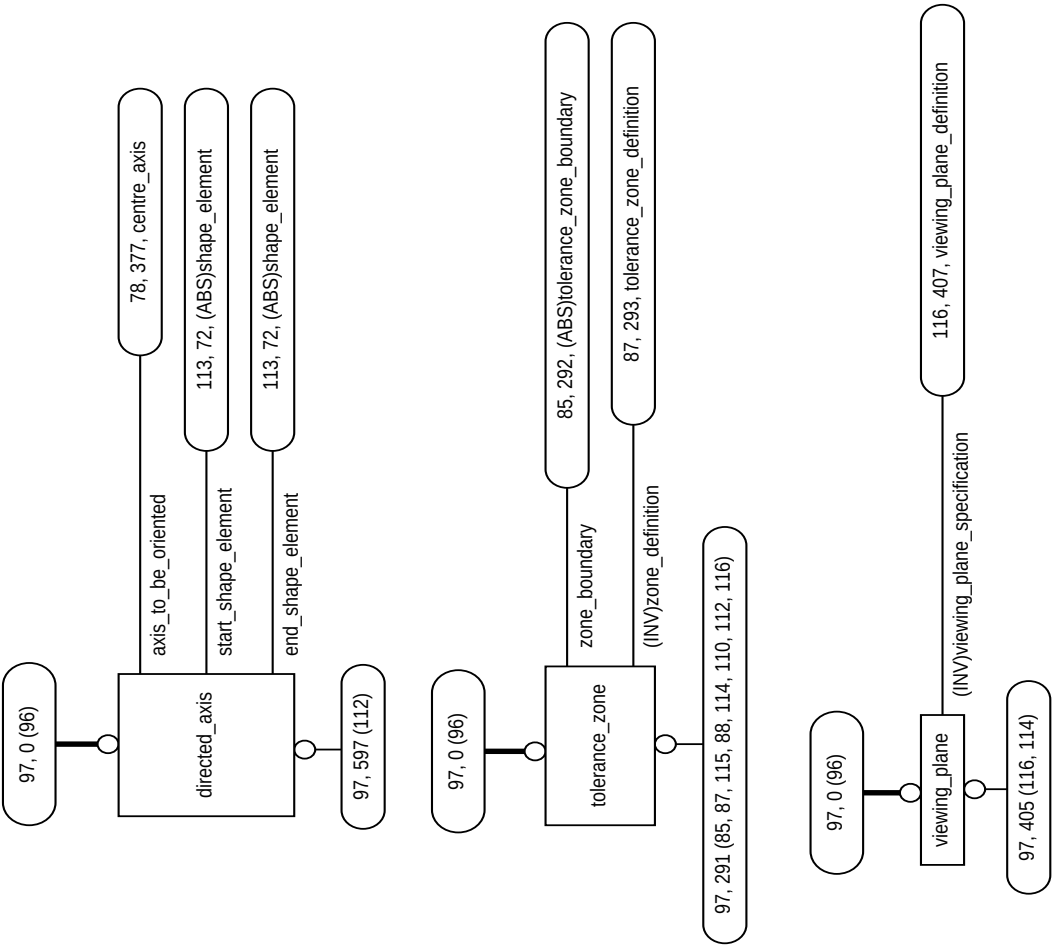


Figure H.97 – ARM EXPRESS-G Graphic 97 of 243

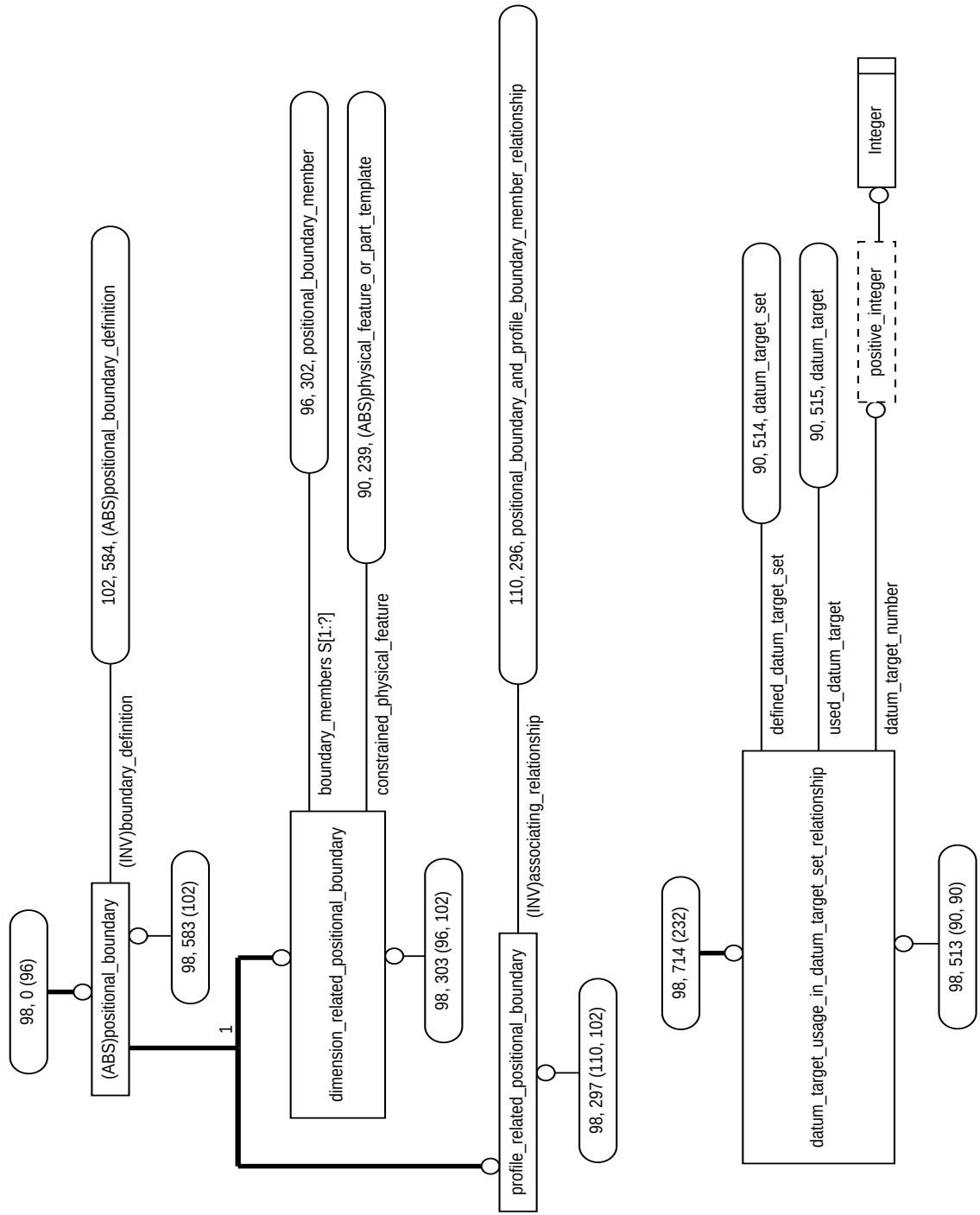


Figure H.98 – ARM EXPRESS-G Graphic 98 of 243

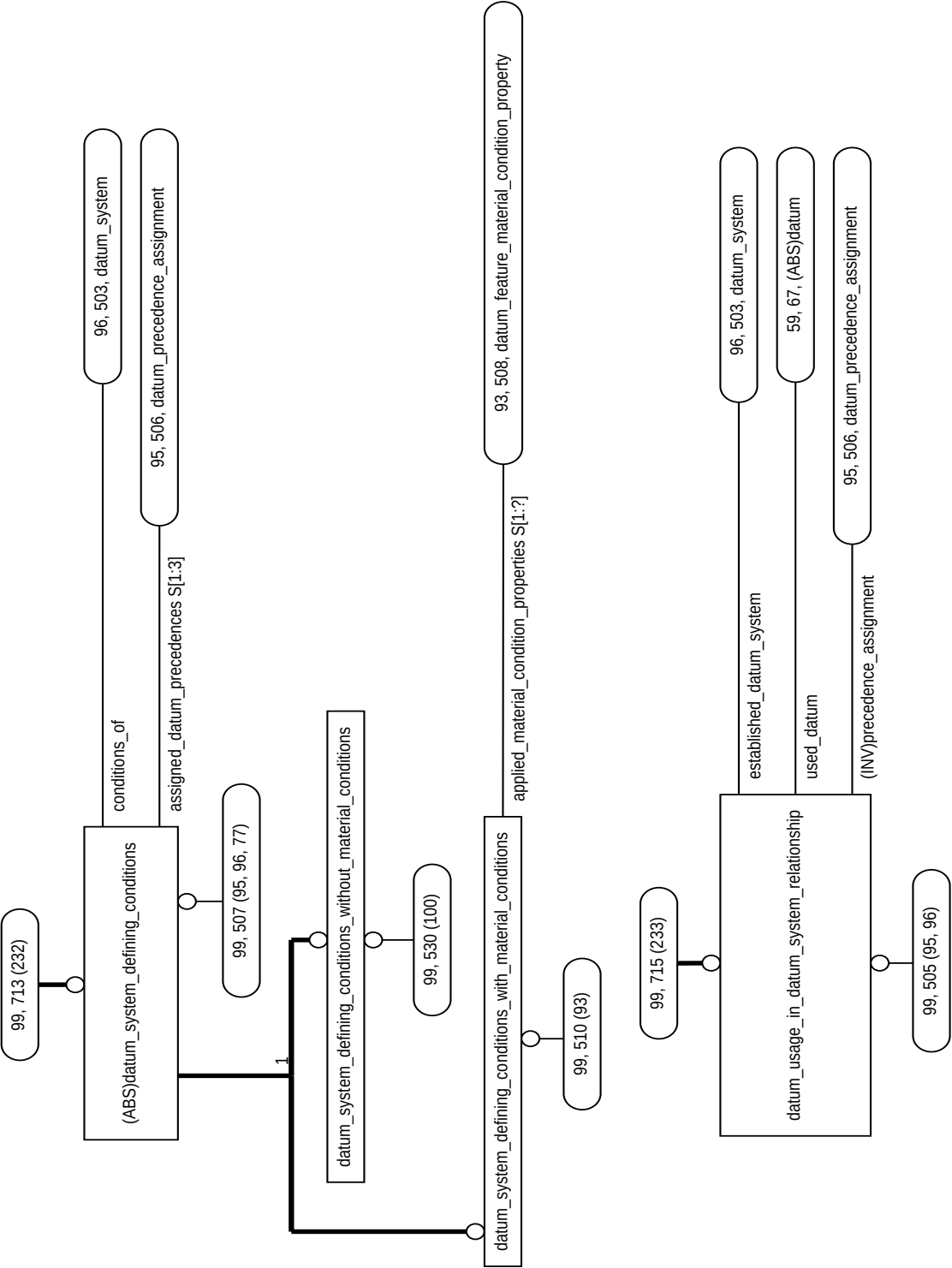


Figure H.99 – ARM EXPRESS-G Graphic 99 of 243

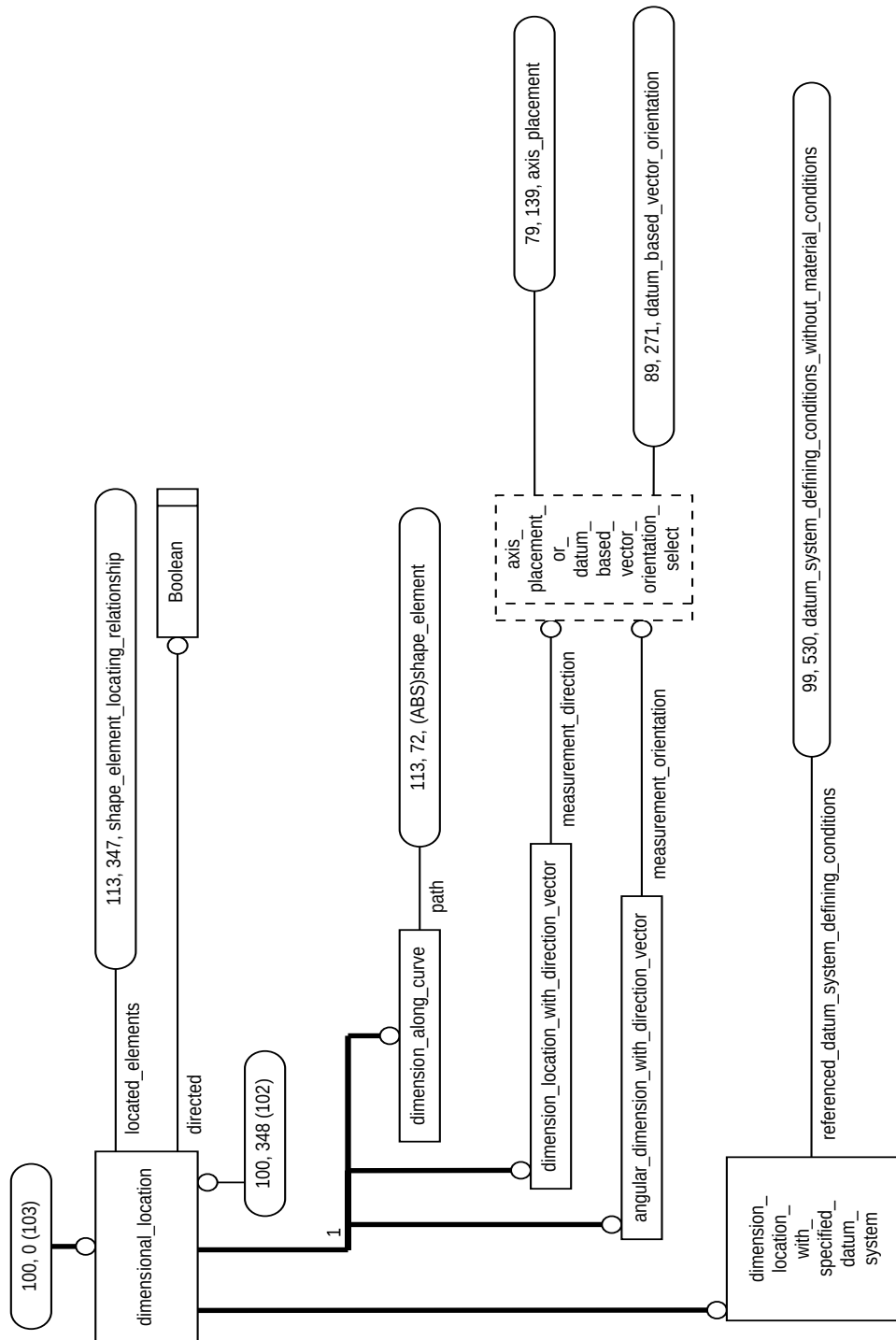


Figure H.100 – ARM EXPRESS-G Graphic 100 of 243

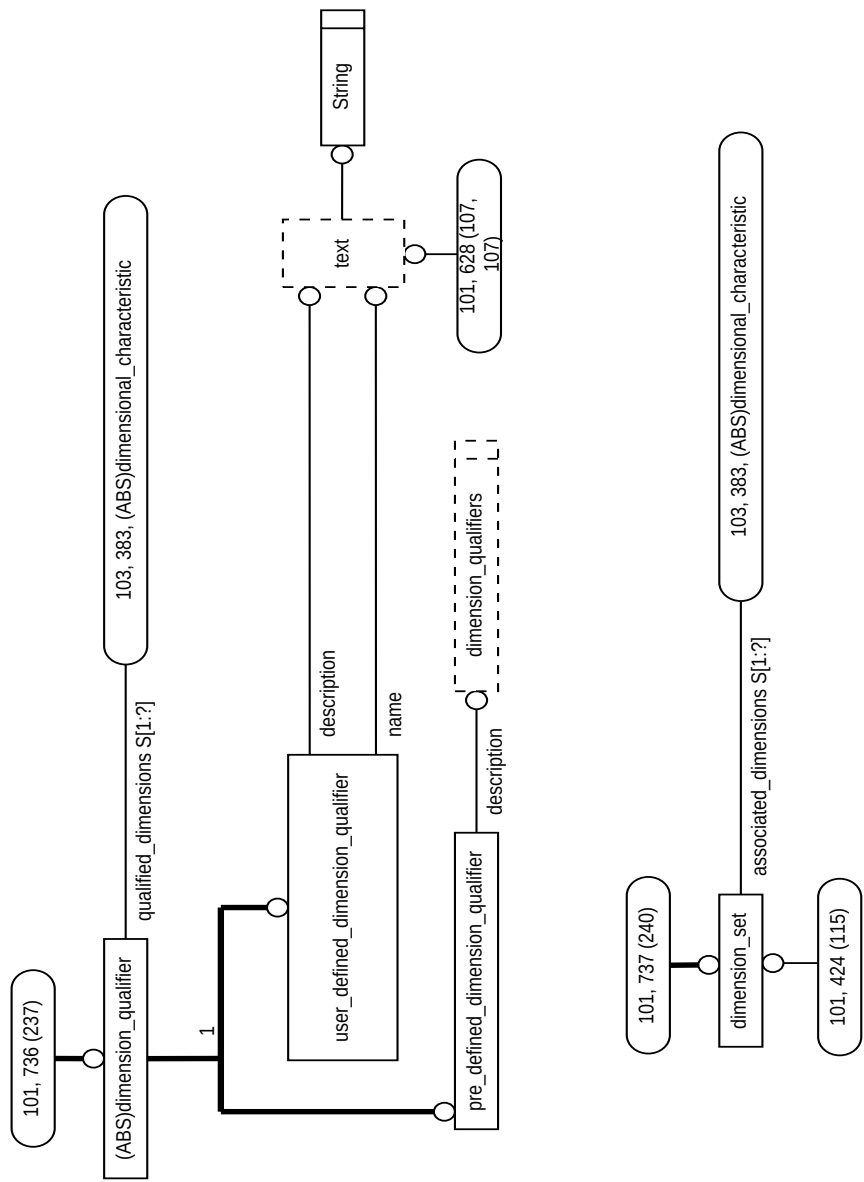


Figure H.101 – ARM EXPRESS-G Graphic 101 of 243

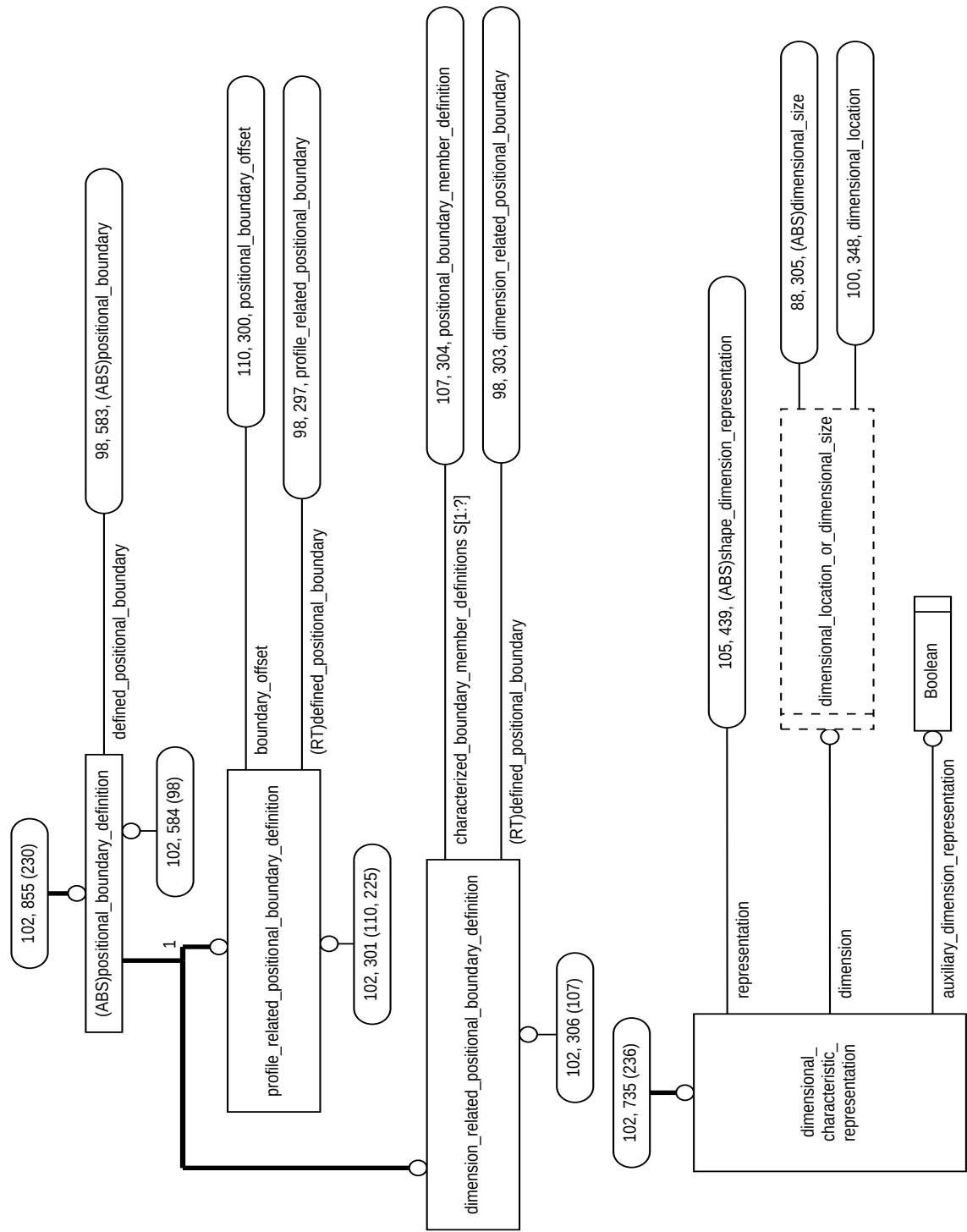


Figure H.102 – ARM EXPRESS-G Graphic 102 of 243

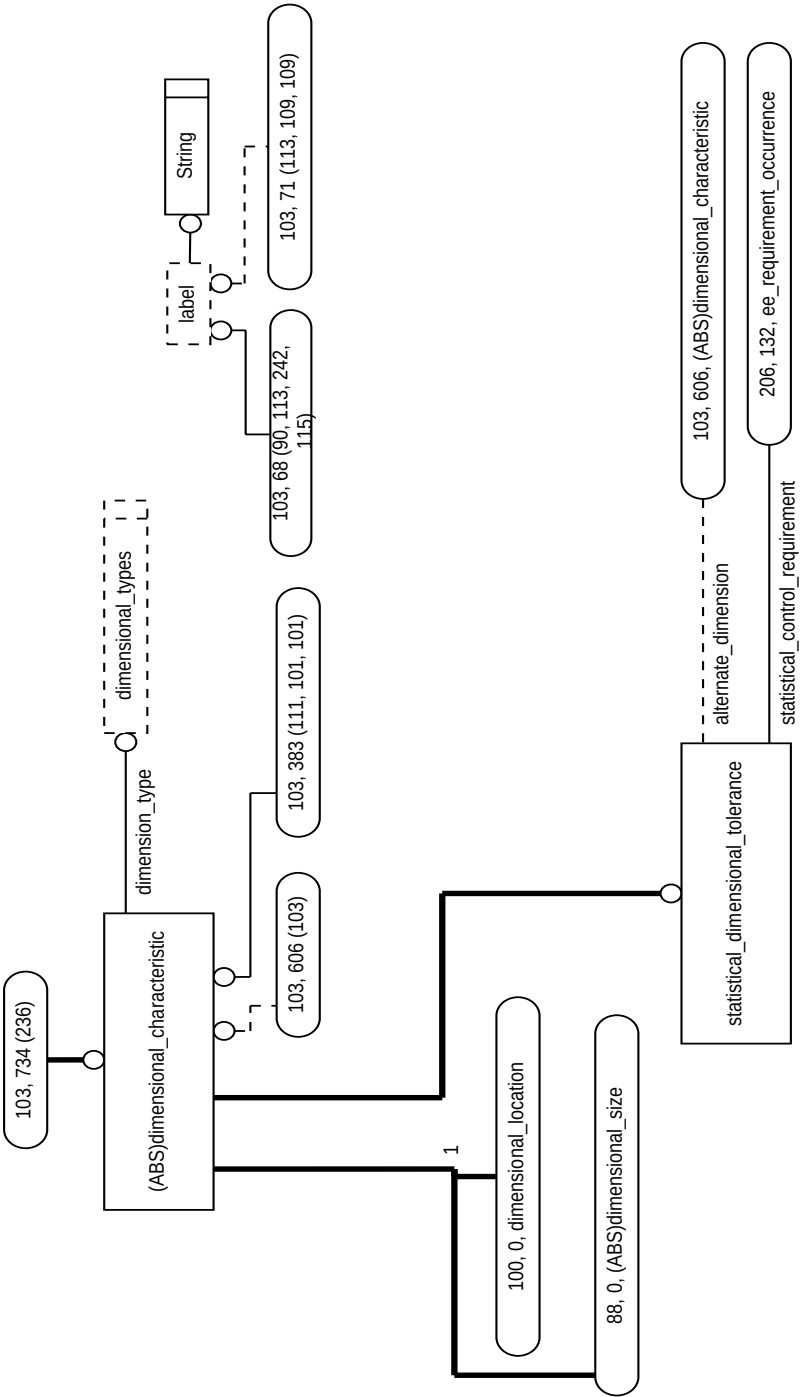


Figure H.103 – ARM EXPRESS-G Graphic 103 of 243

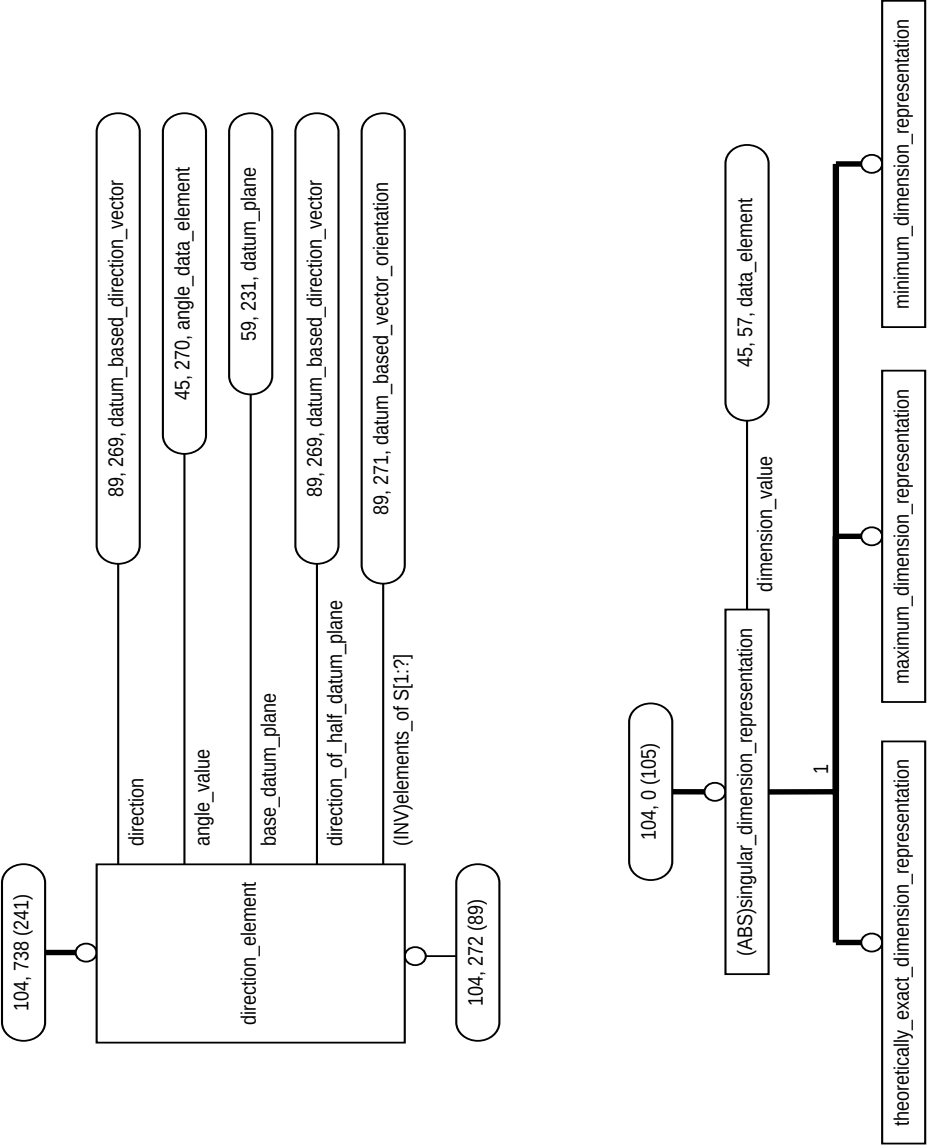


Figure H.104 – ARM EXPRESS-G Graphic 104 of 243

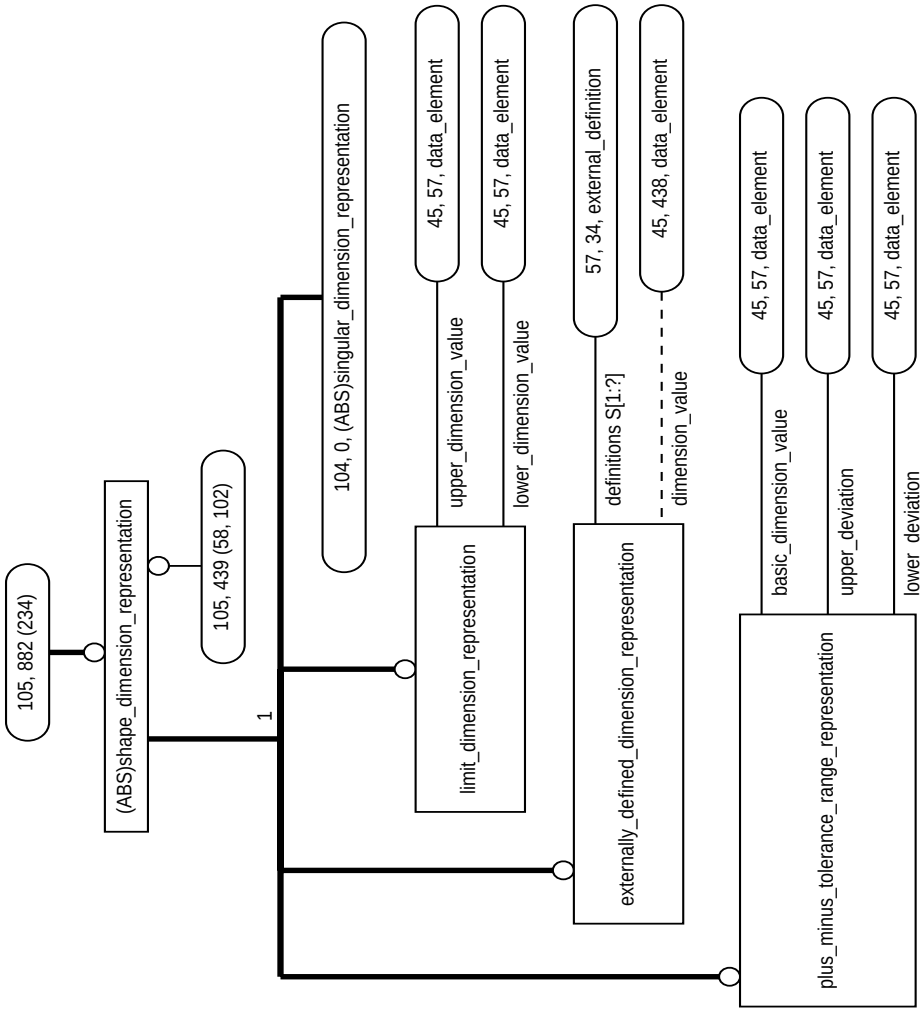


Figure H.105 – ARM EXPRESS-G Graphic 105 of 243

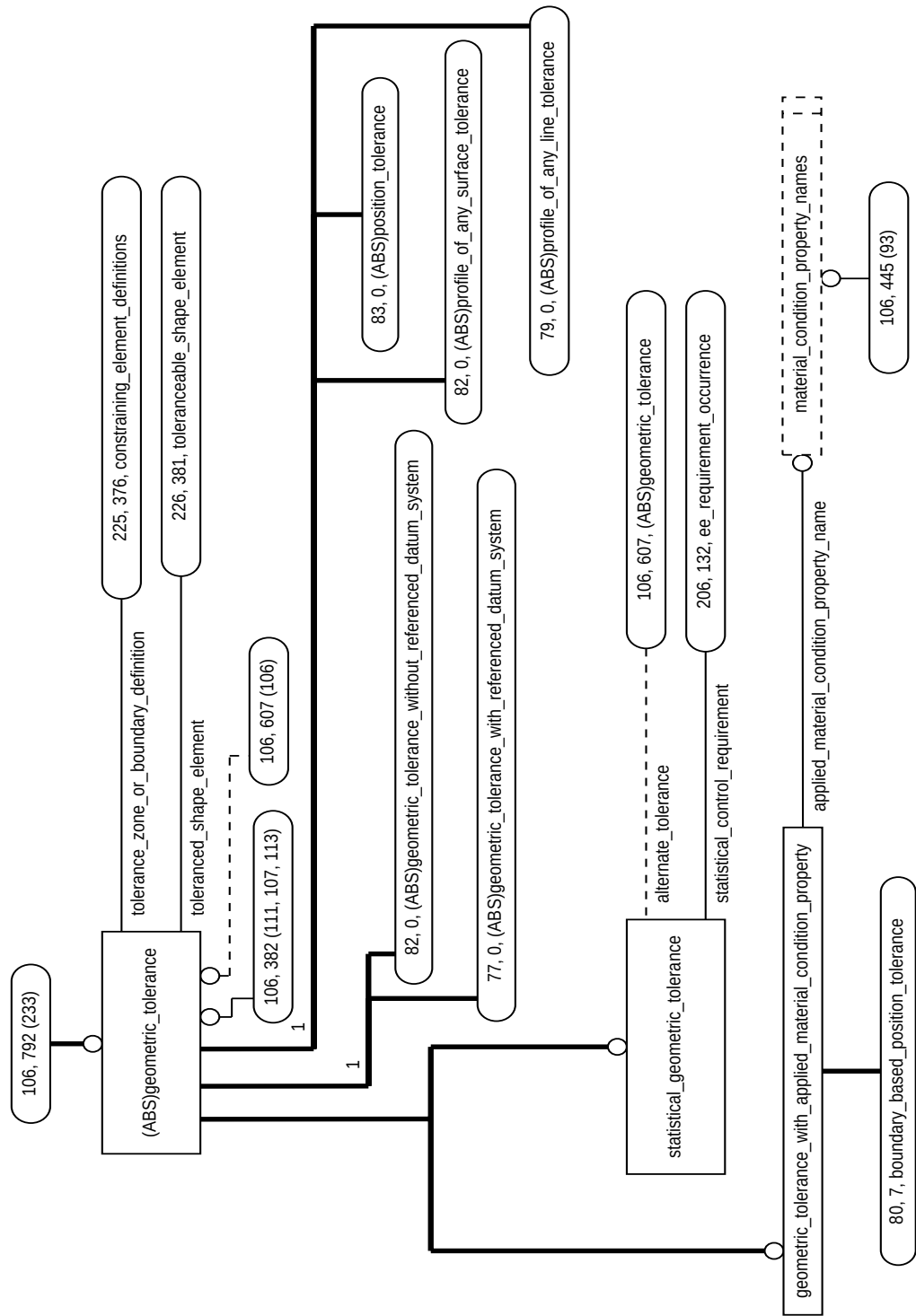


Figure H.106 – ARM EXPRESS-G Graphic 106 of 243

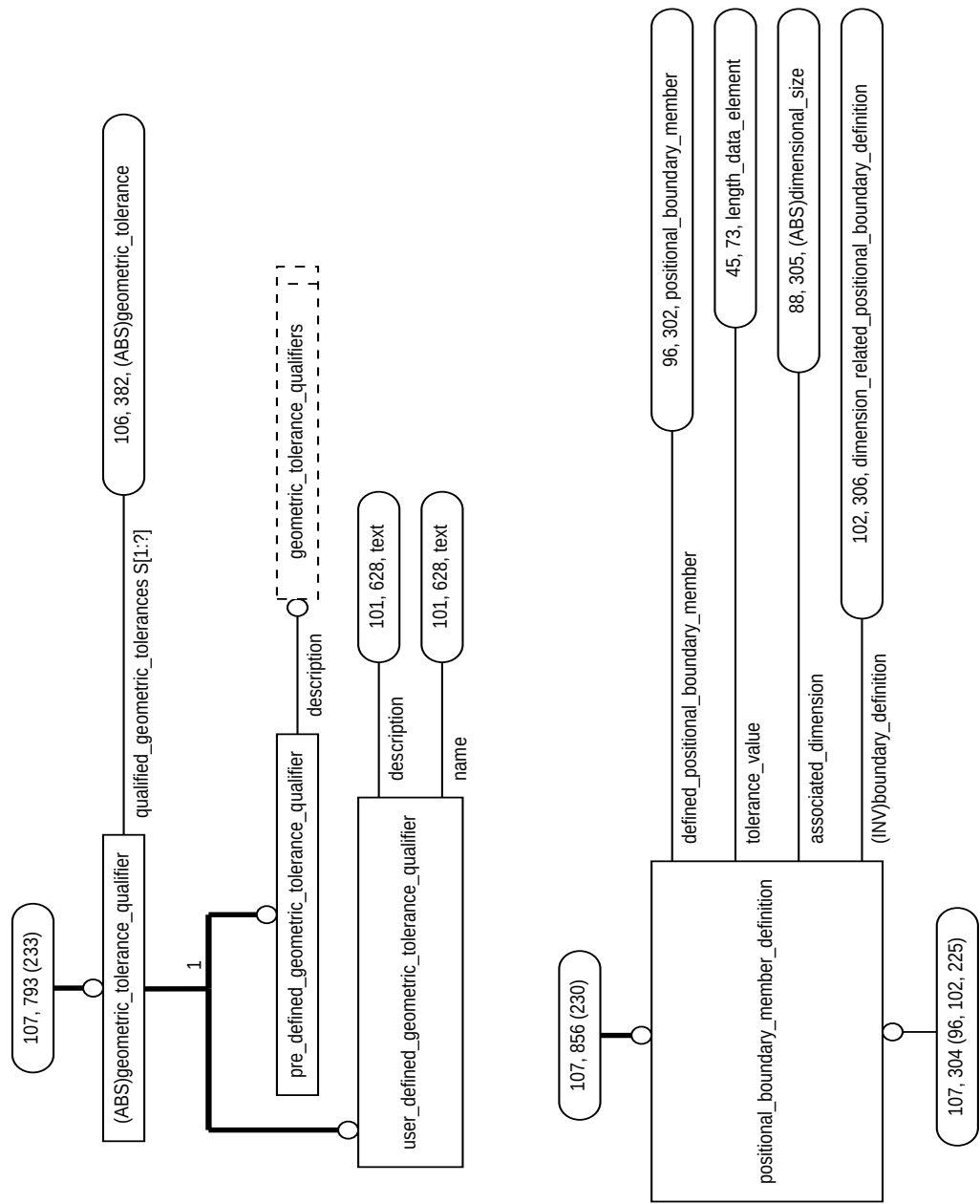


Figure H.107 – ARM EXPRESS-G Graphic 107 of 243

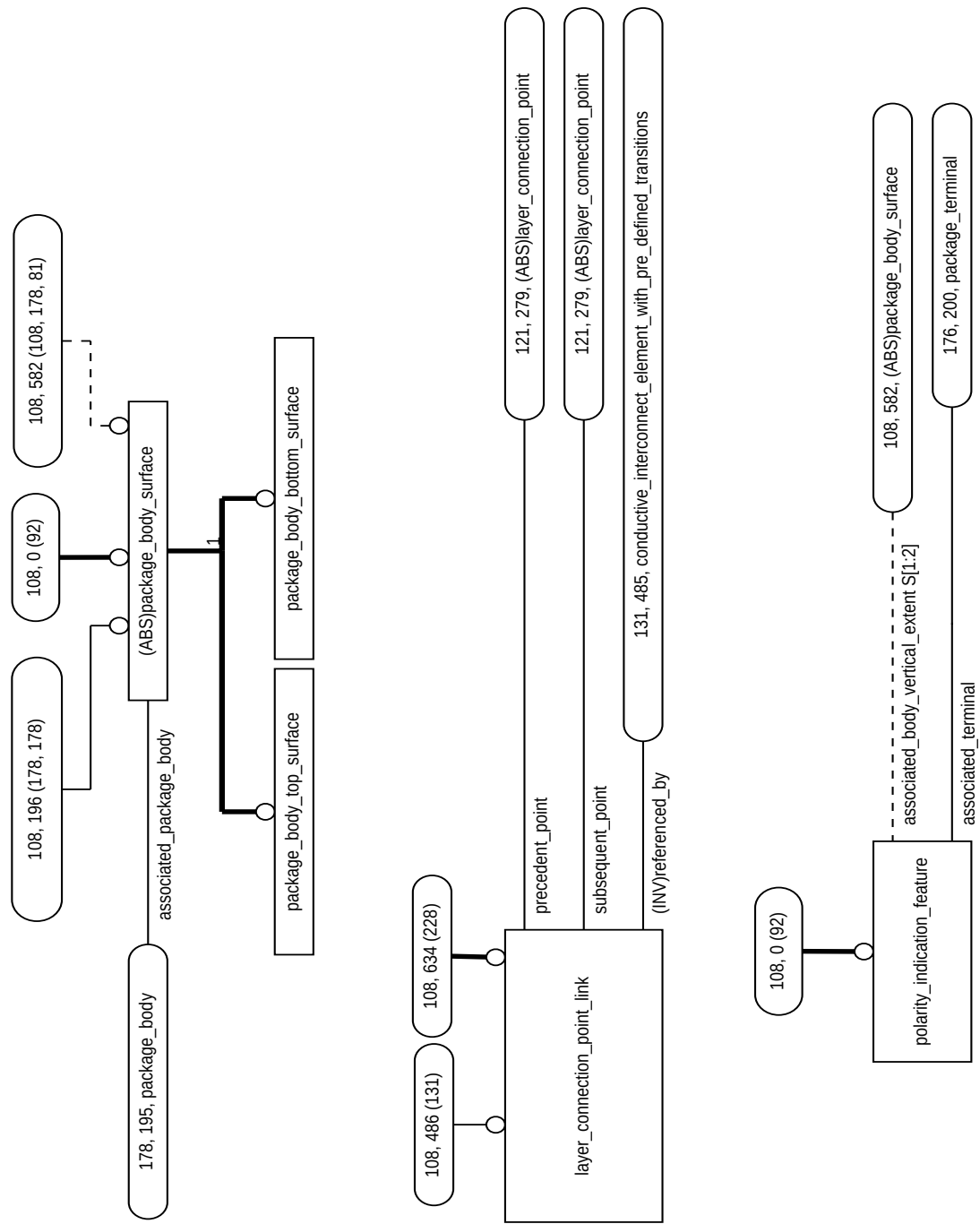


Figure H.108 – ARM EXPRESS-G Graphic 108 of 243

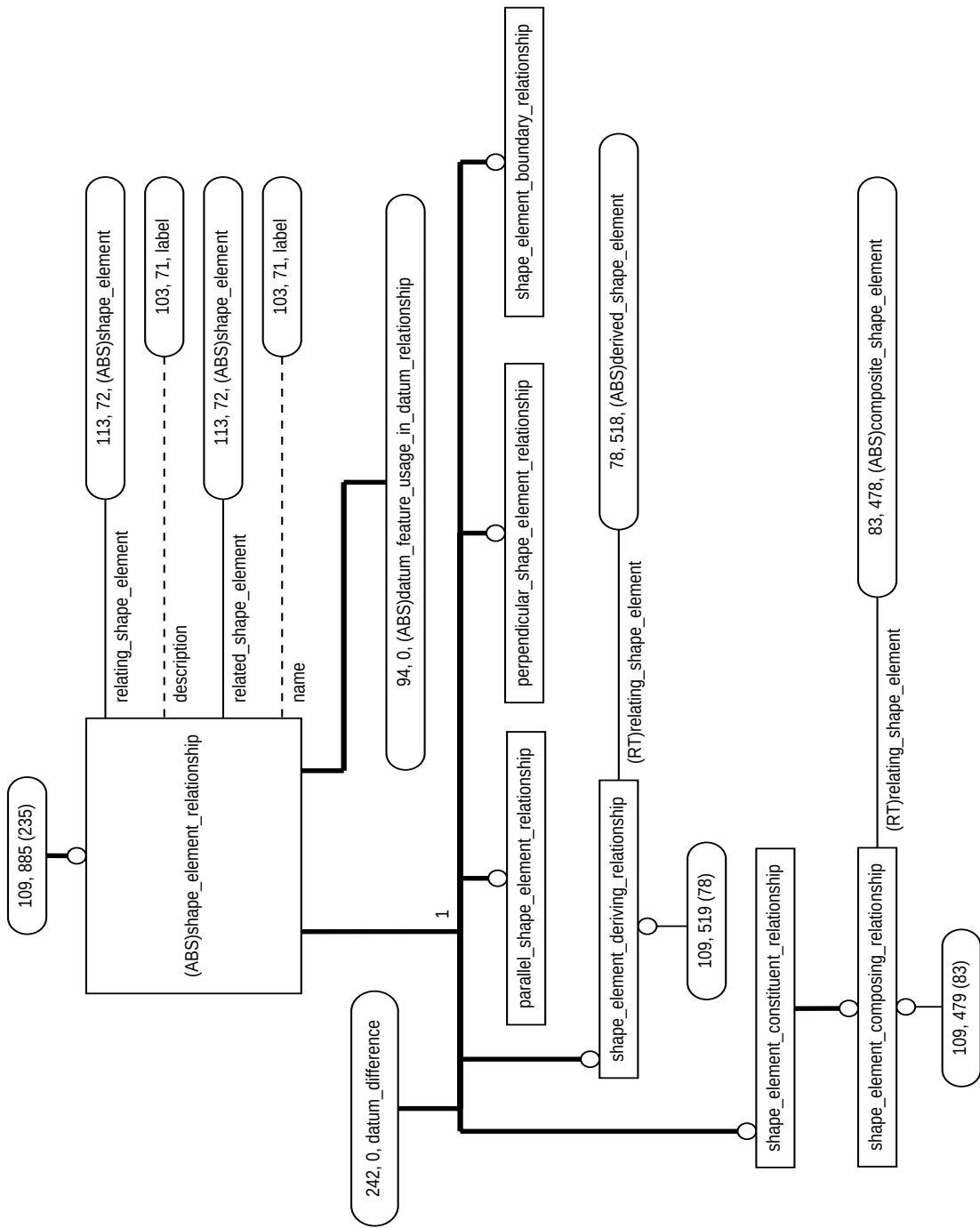


Figure H.109 – ARM EXPRESS-G Graphic 109 of 243

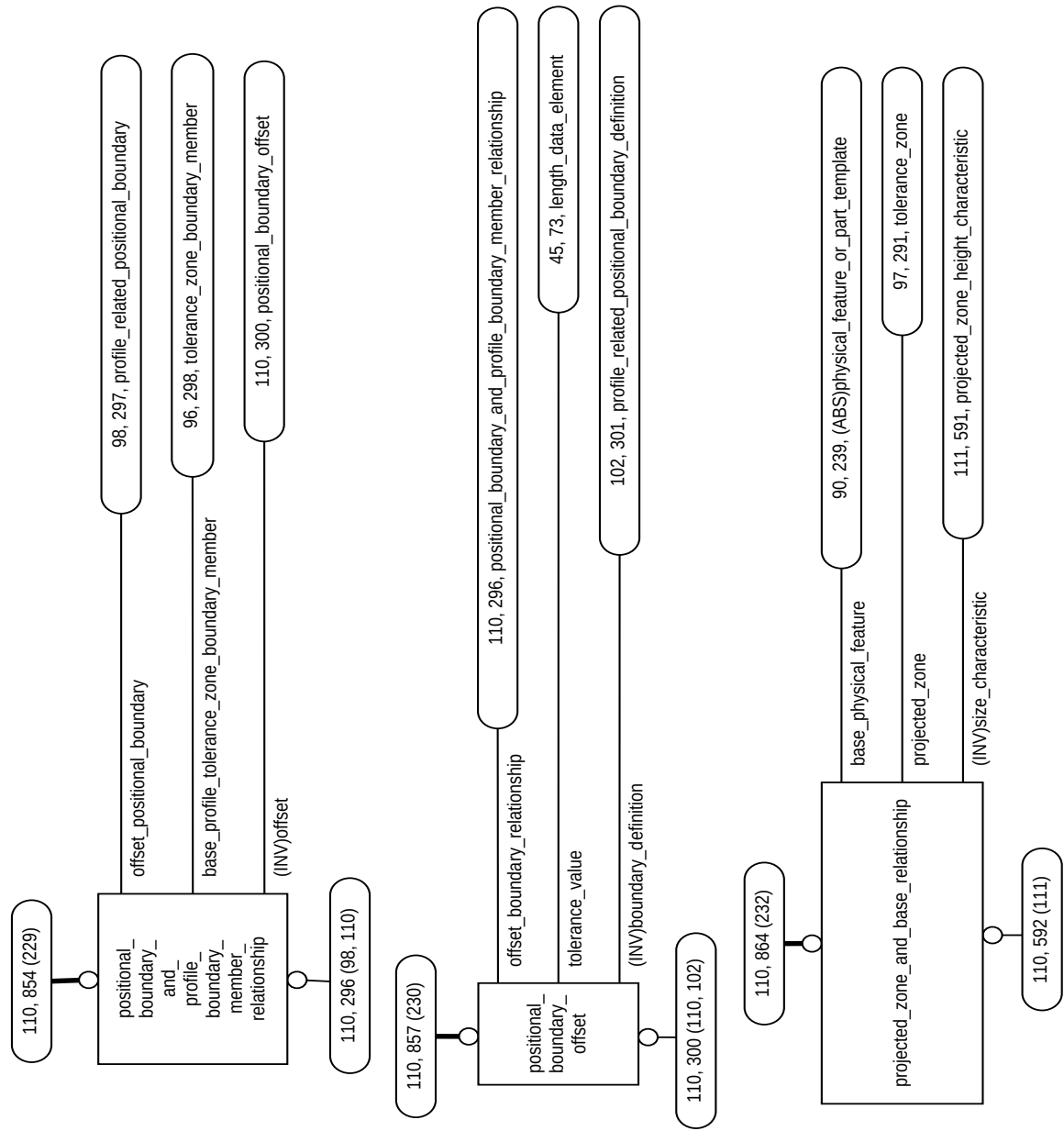


Figure H.110 – ARM EXPRESS-G Graphic 110 of 243

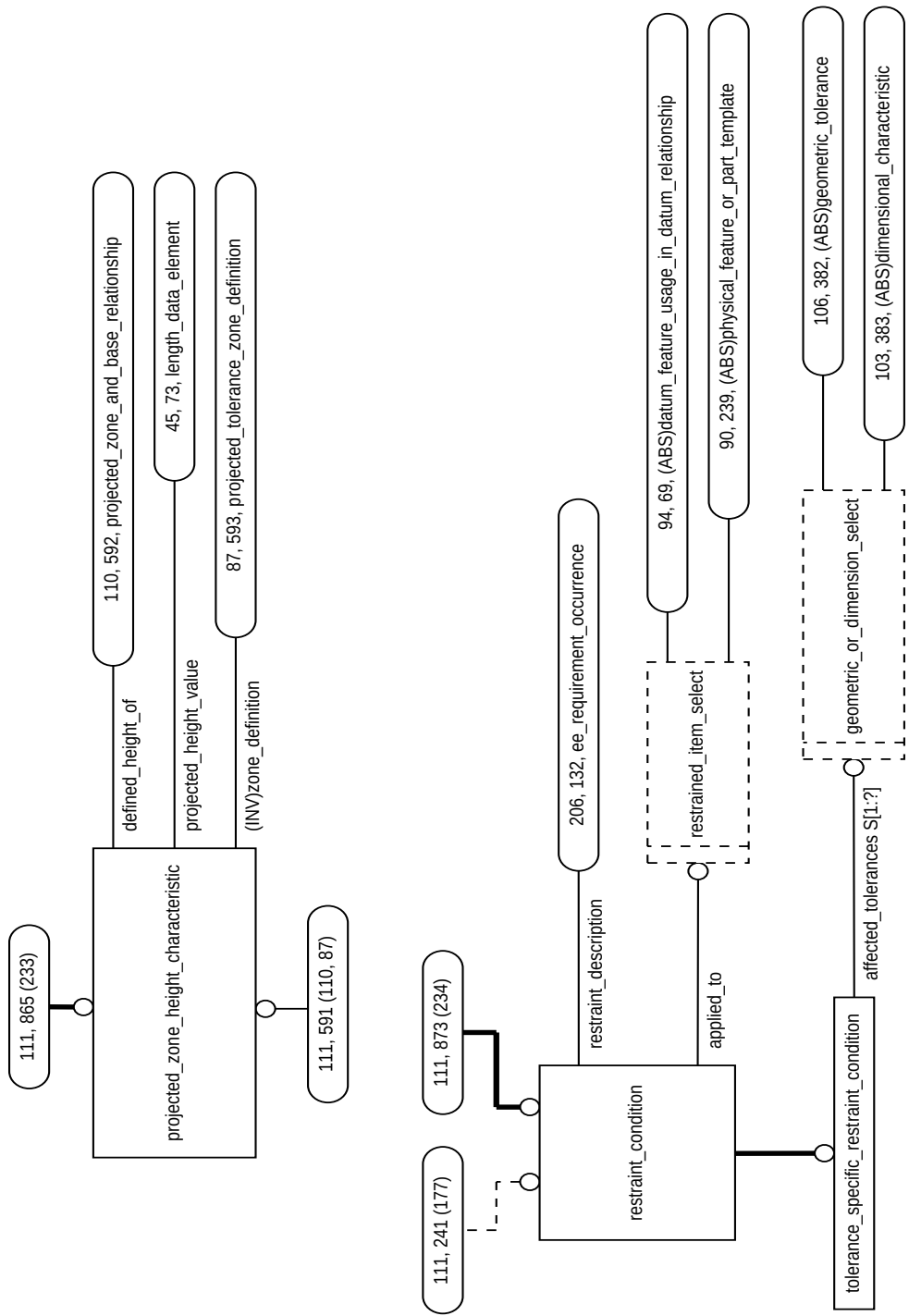


Figure H.111 – ARM EXPRESS-G Graphic 111 of 243

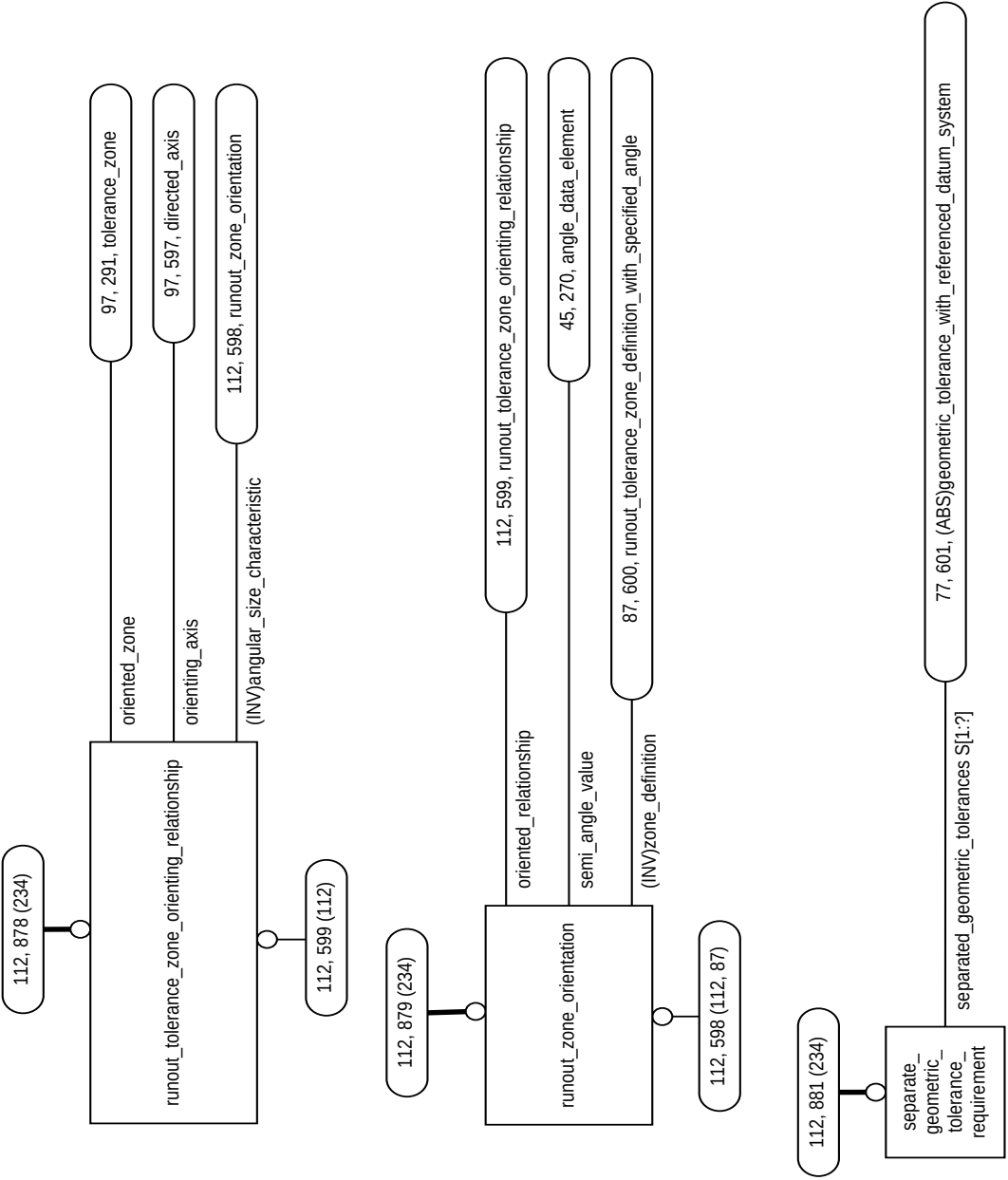


Figure H.112 – ARM EXPRESS-G Graphic 112 of 243

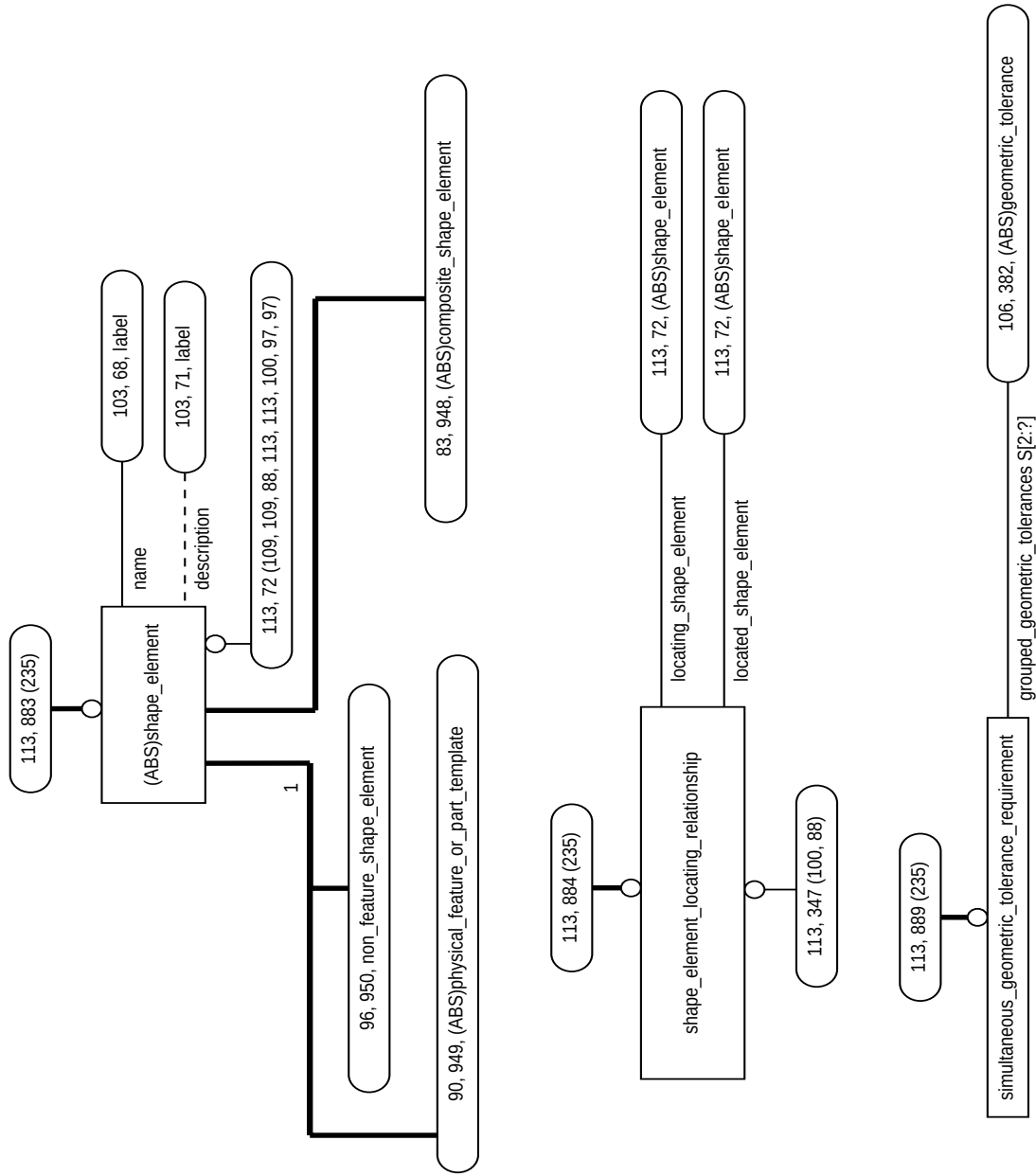


Figure H.113 – ARM EXPRESS-G Graphic 113 of 243

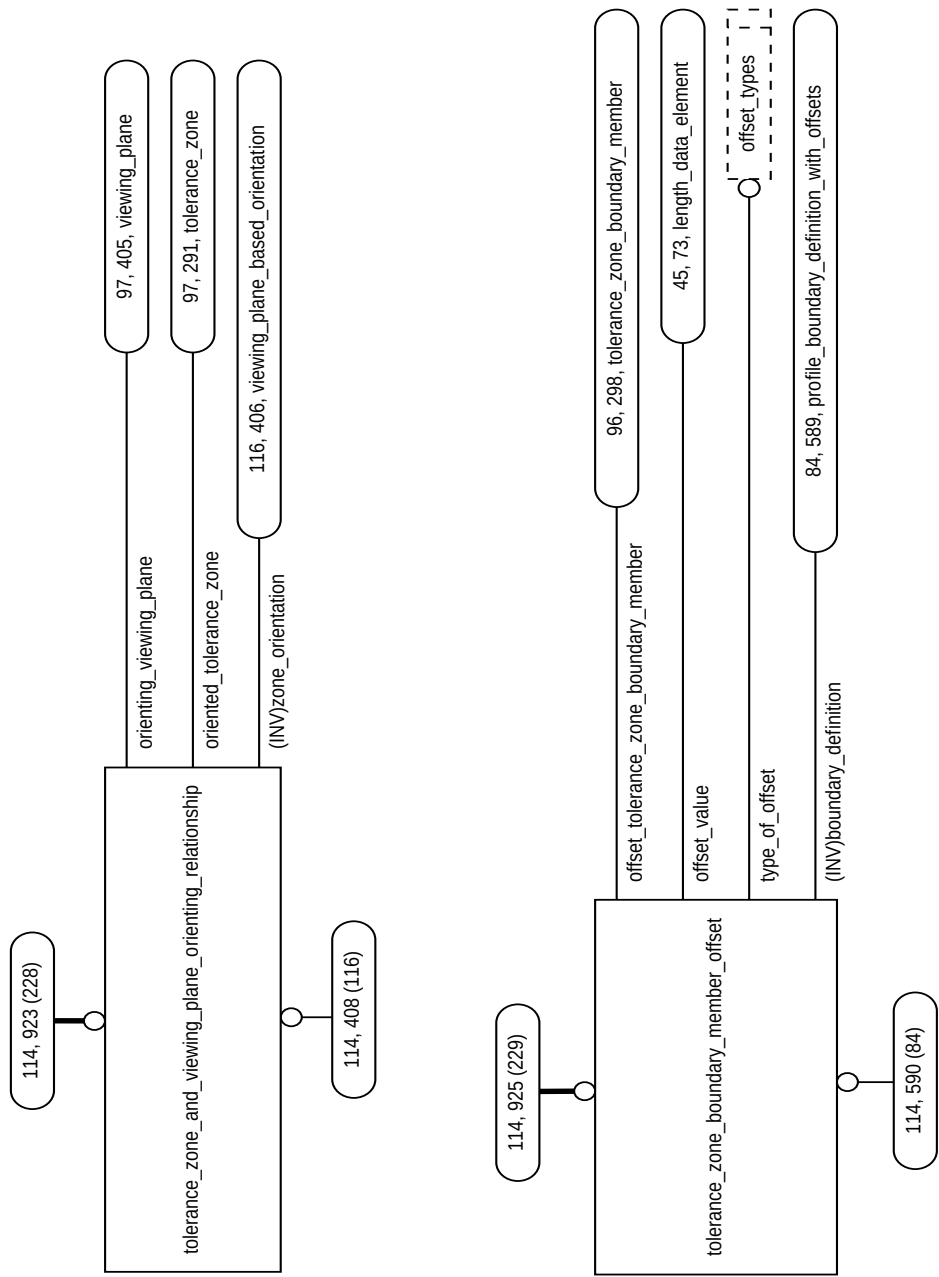


Figure H.114 – ARM EXPRESS-G Graphic 114 of 243

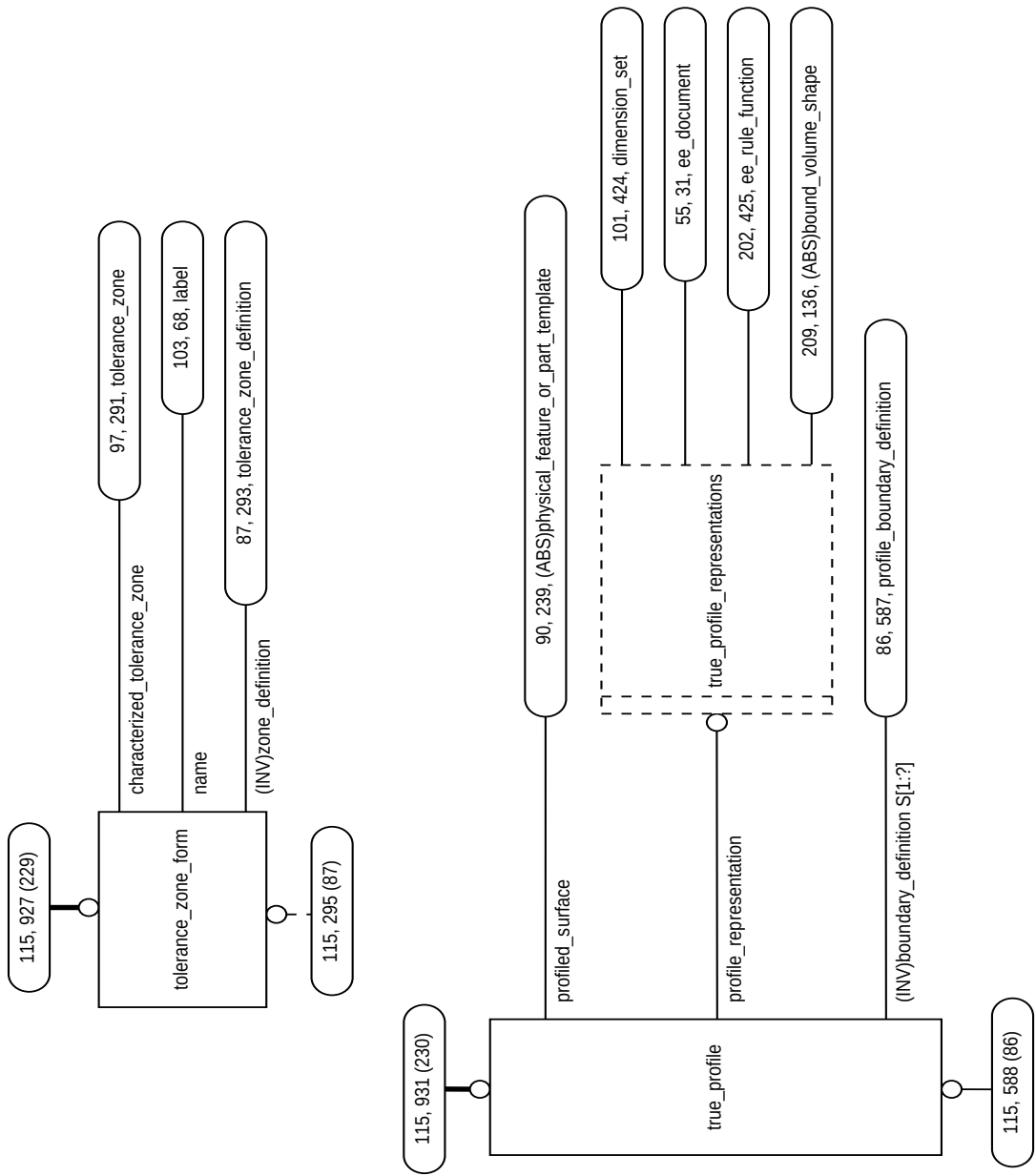


Figure H.115 – ARM EXPRESS-G Graphic 115 of 243

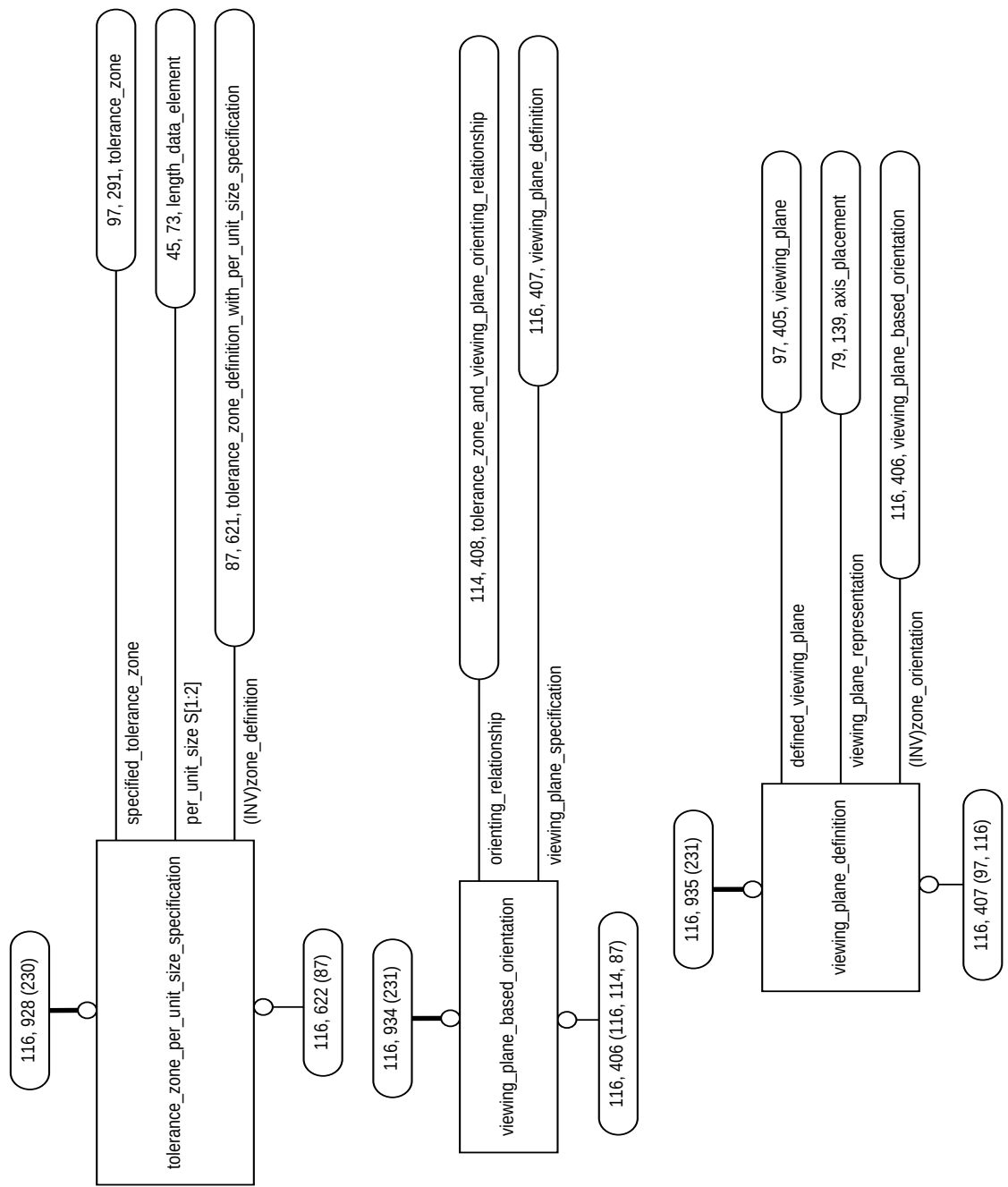


Figure H.116 – ARM EXPRESS-G Graphic 116 of 243

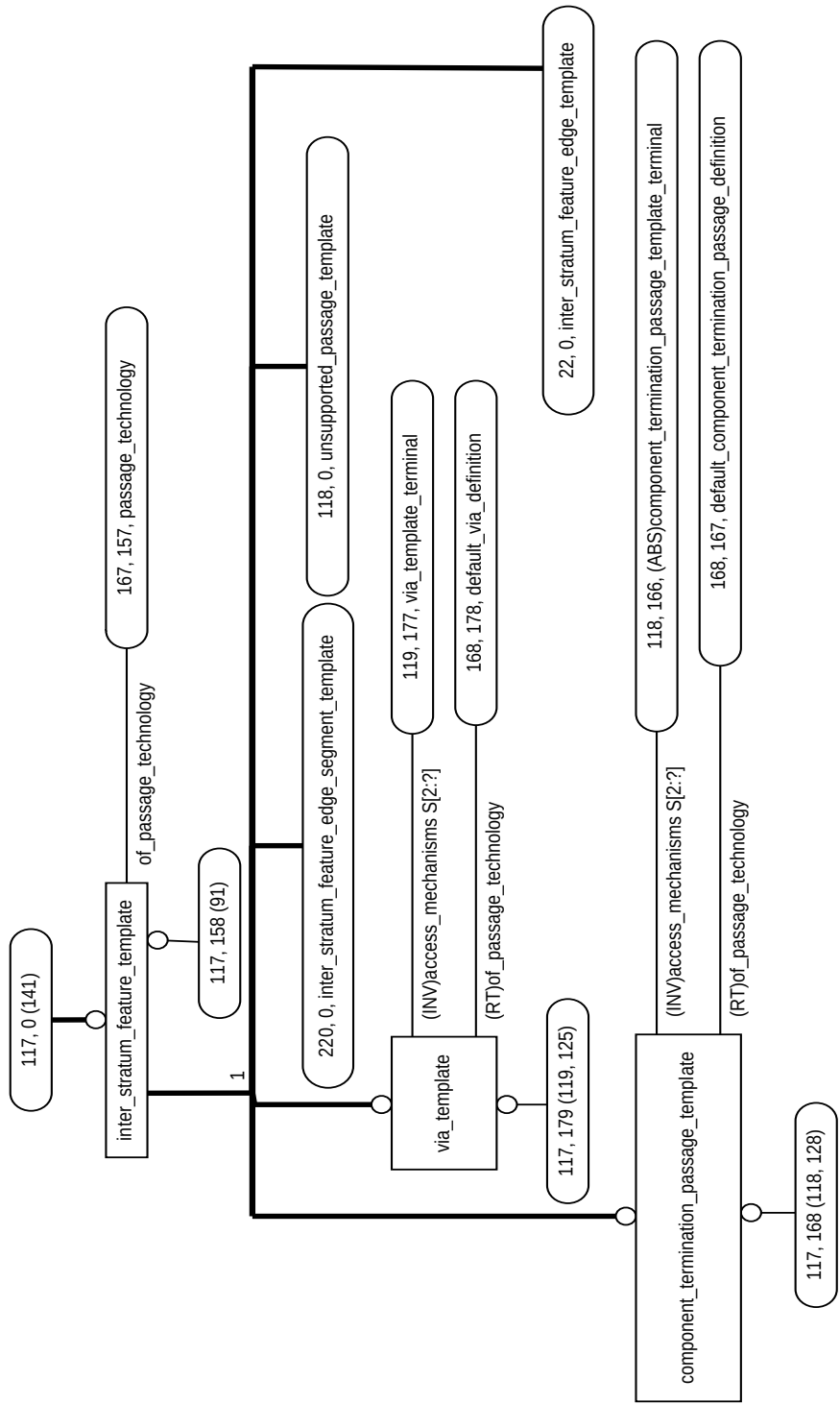


Figure H.117 – ARM EXPRESS-G Graphic 117 of 243

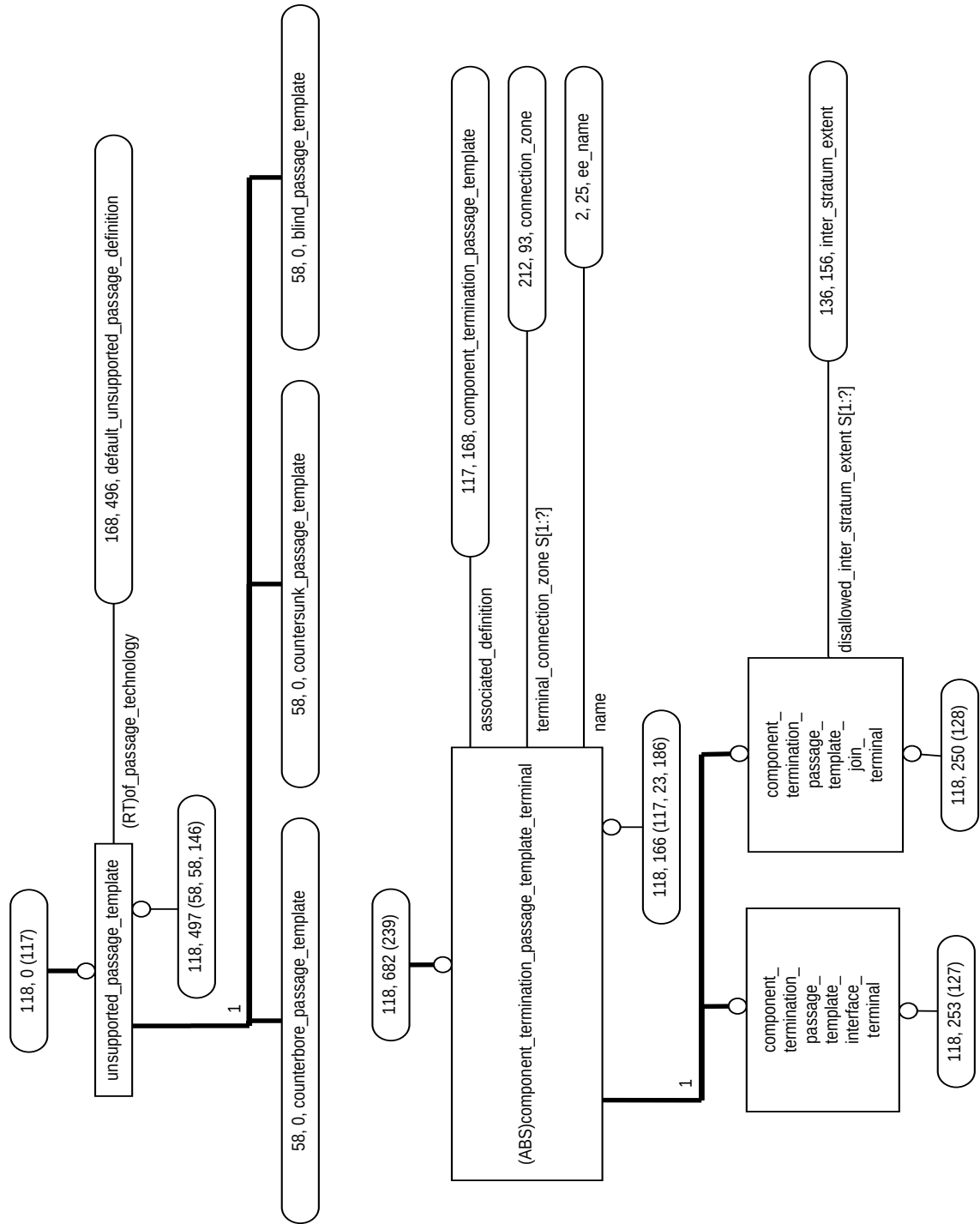


Figure H.118 – ARM EXPRESS-G Graphic 118 of 243

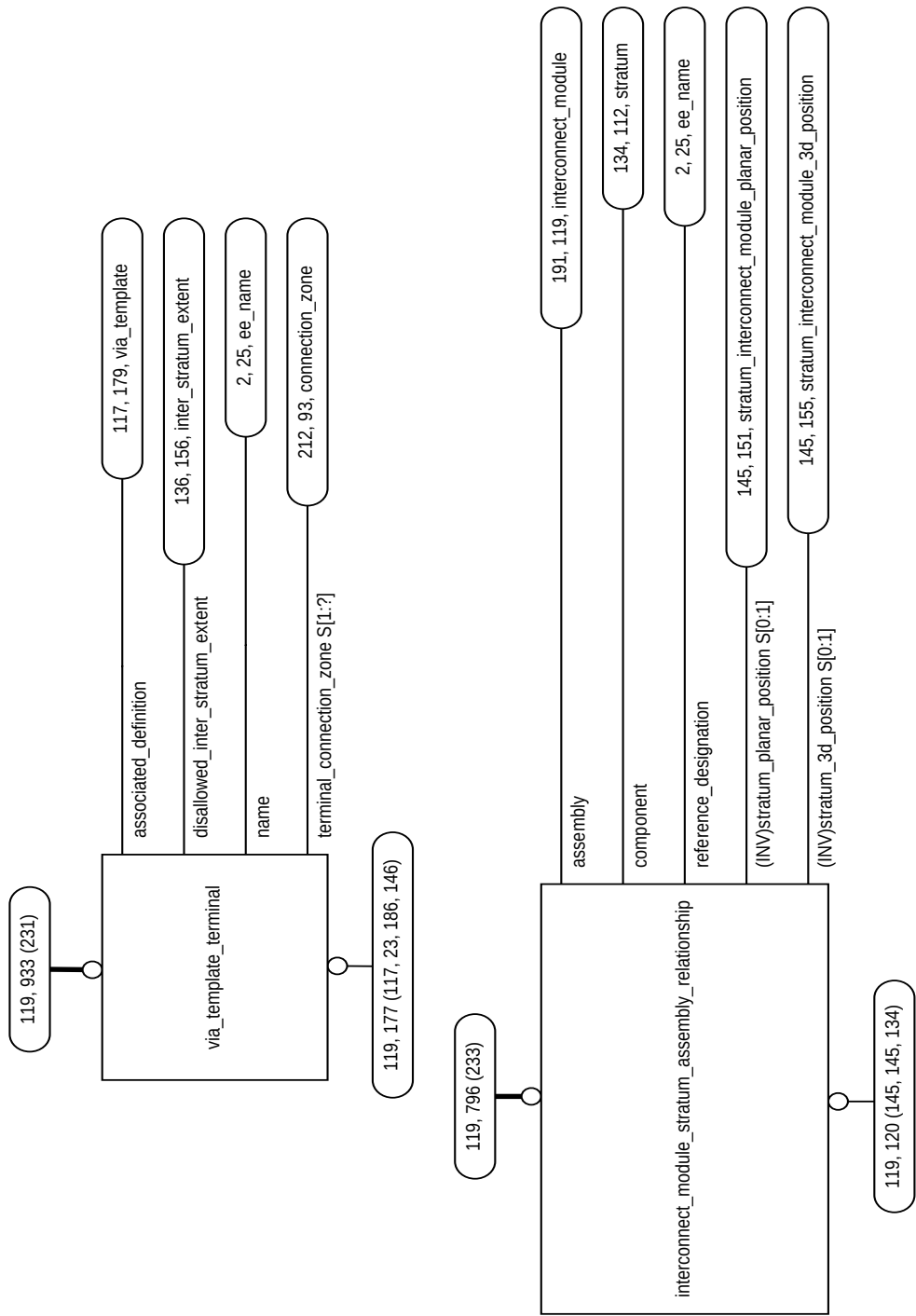


Figure H.119 – ARM EXPRESS-G Graphic 119 of 243

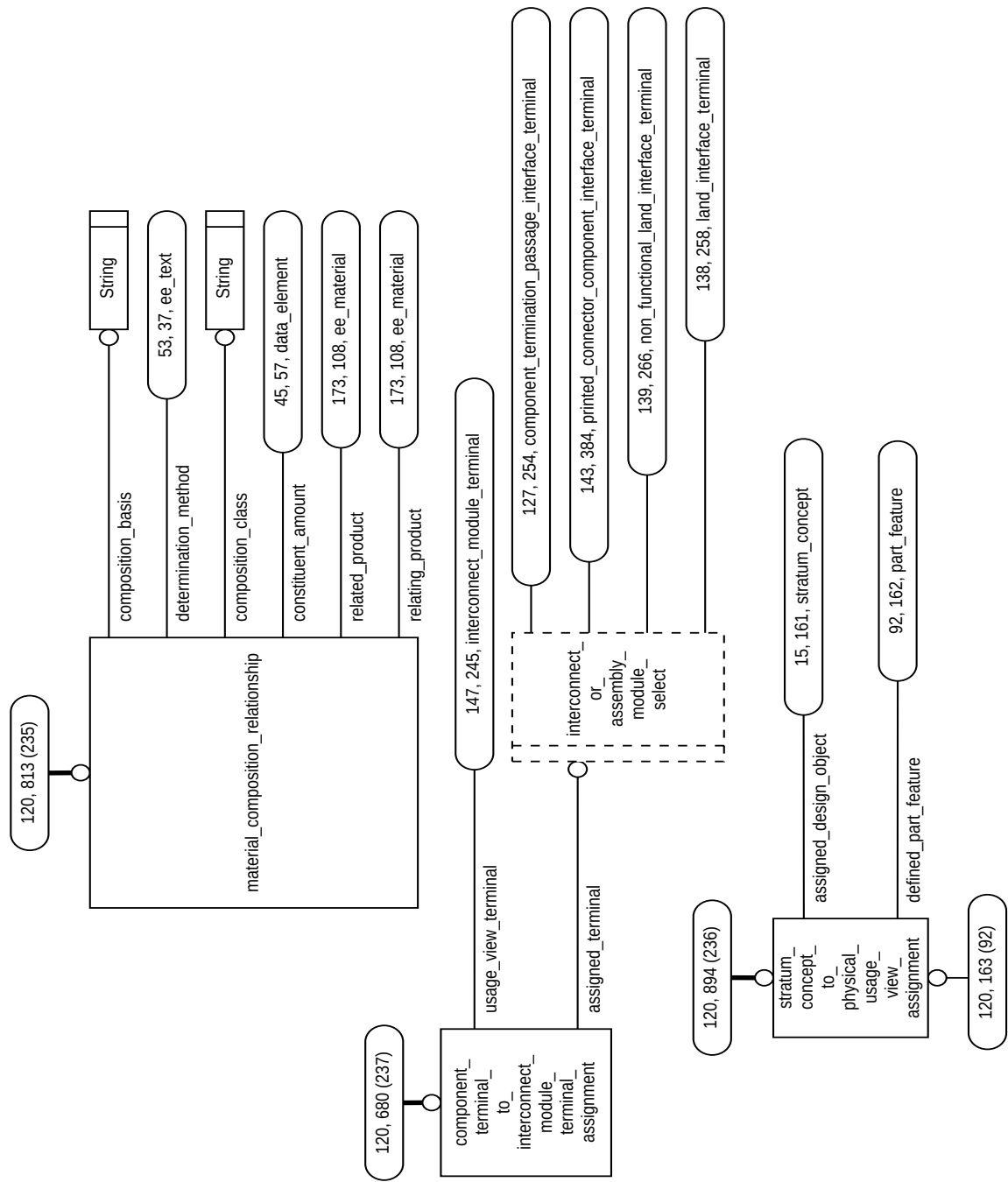


Figure H.120 – ARM EXPRESS-G Graphic 120 of 243

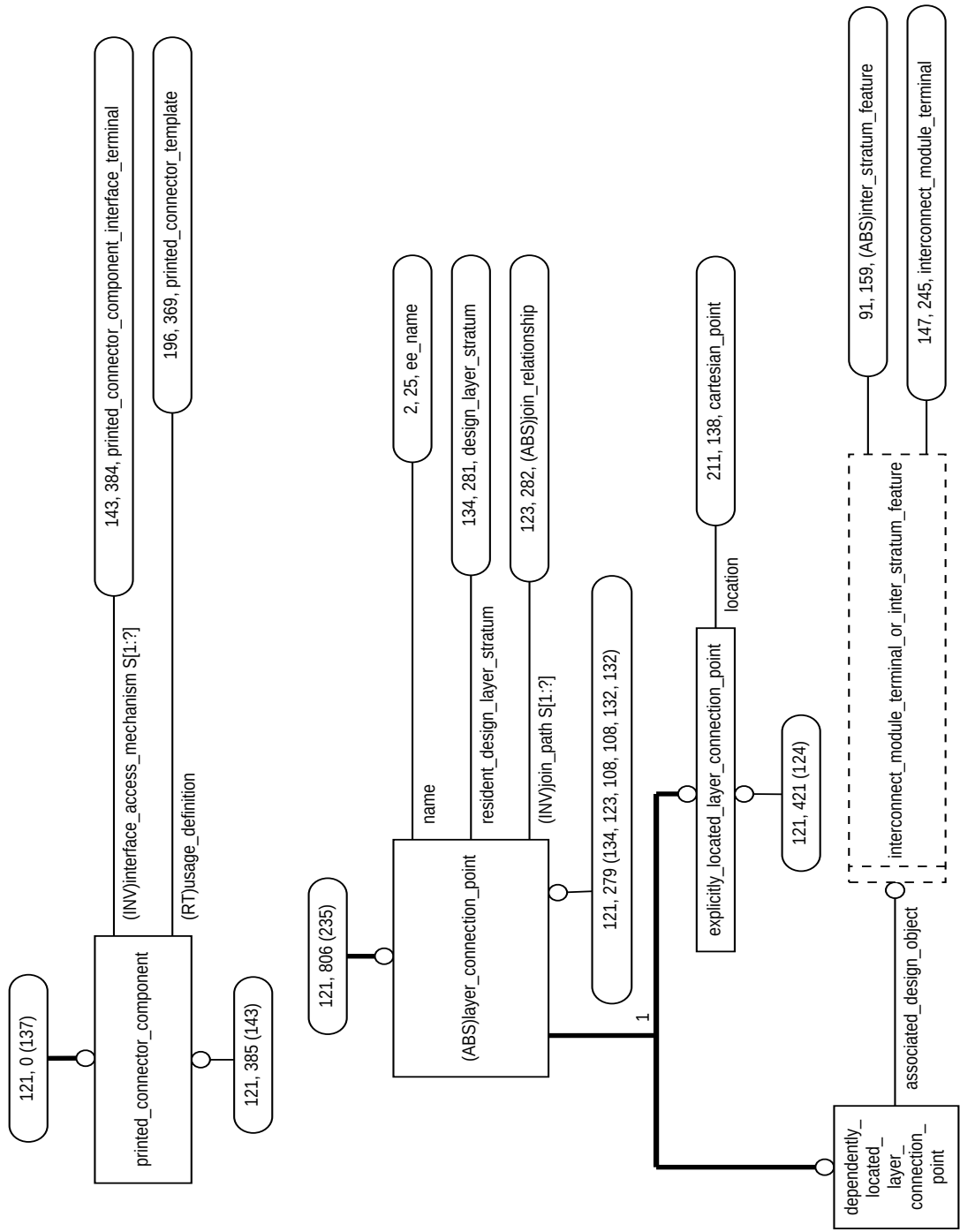


Figure H.121 – ARM EXPRESS-G Graphic 121 of 243

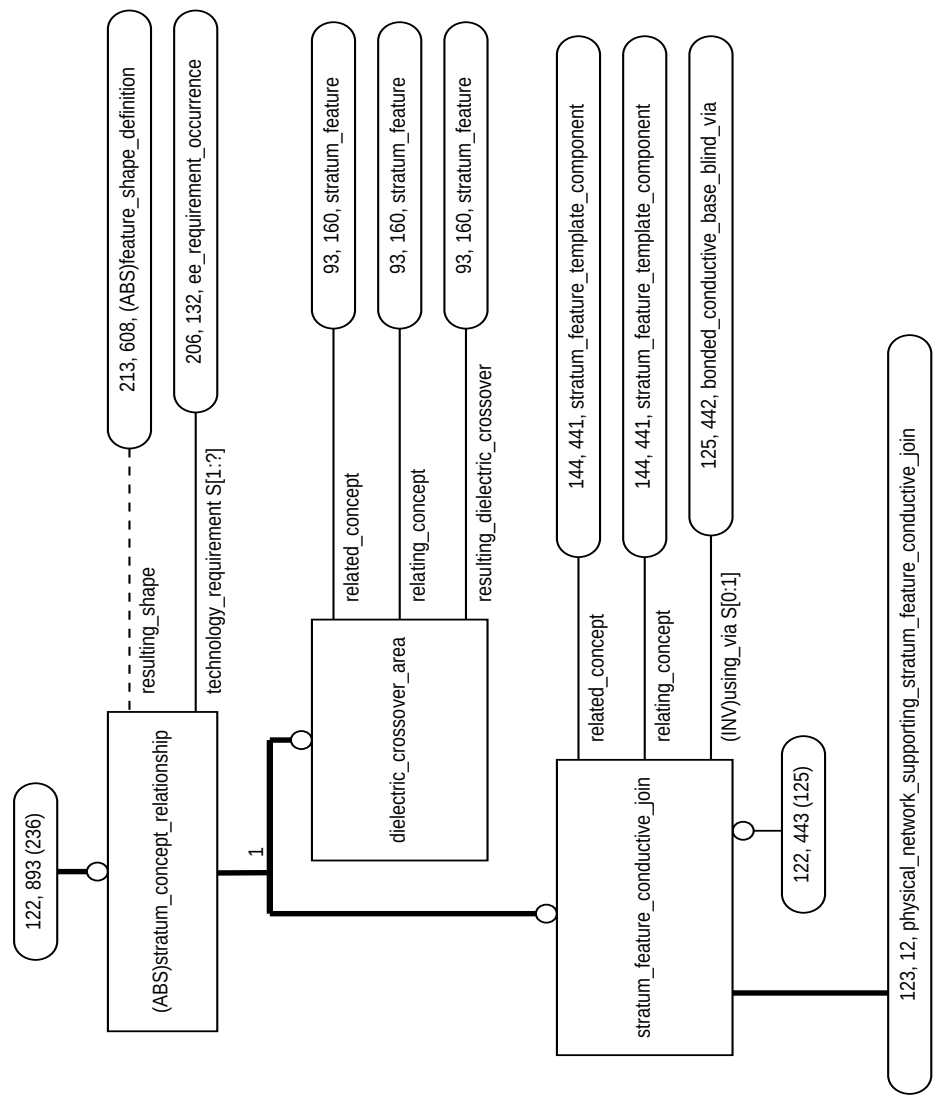


Figure H.122 – ARM EXPRESS-G Graphic 122 of 243

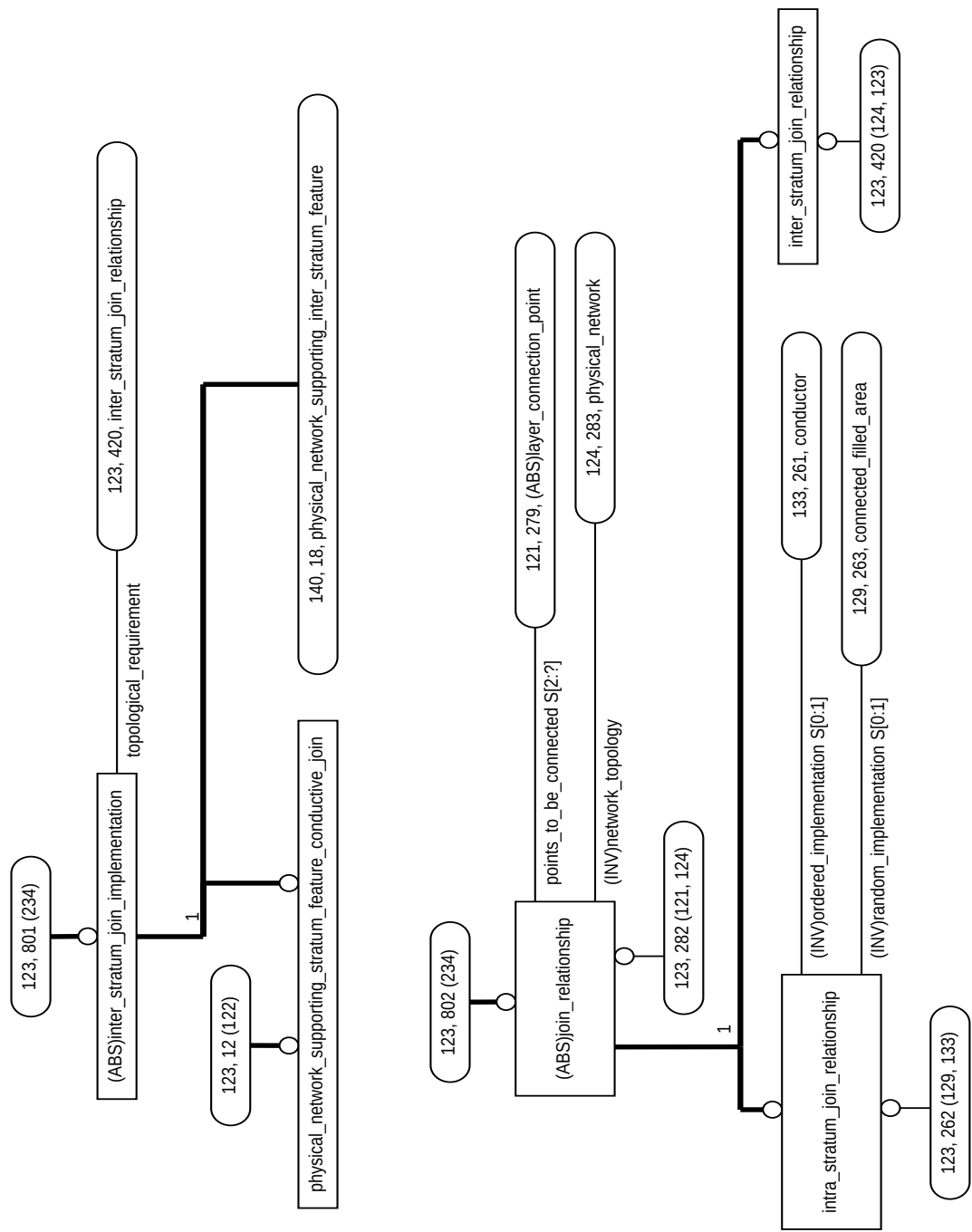


Figure H.123 – ARM EXPRESS-G Graphic 123 of 243

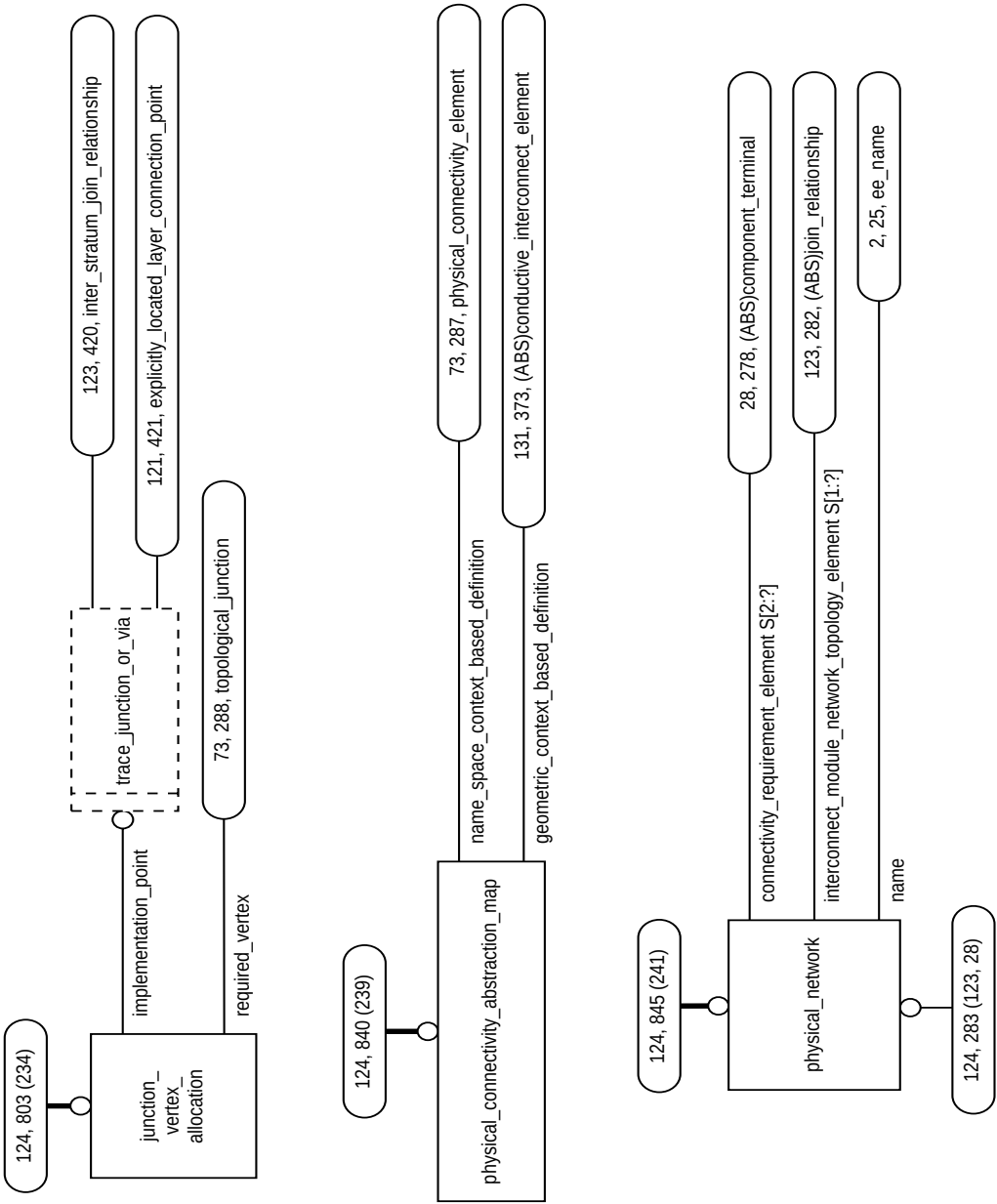


Figure H.124 – ARM EXPRESS-G Graphic 124 of 243

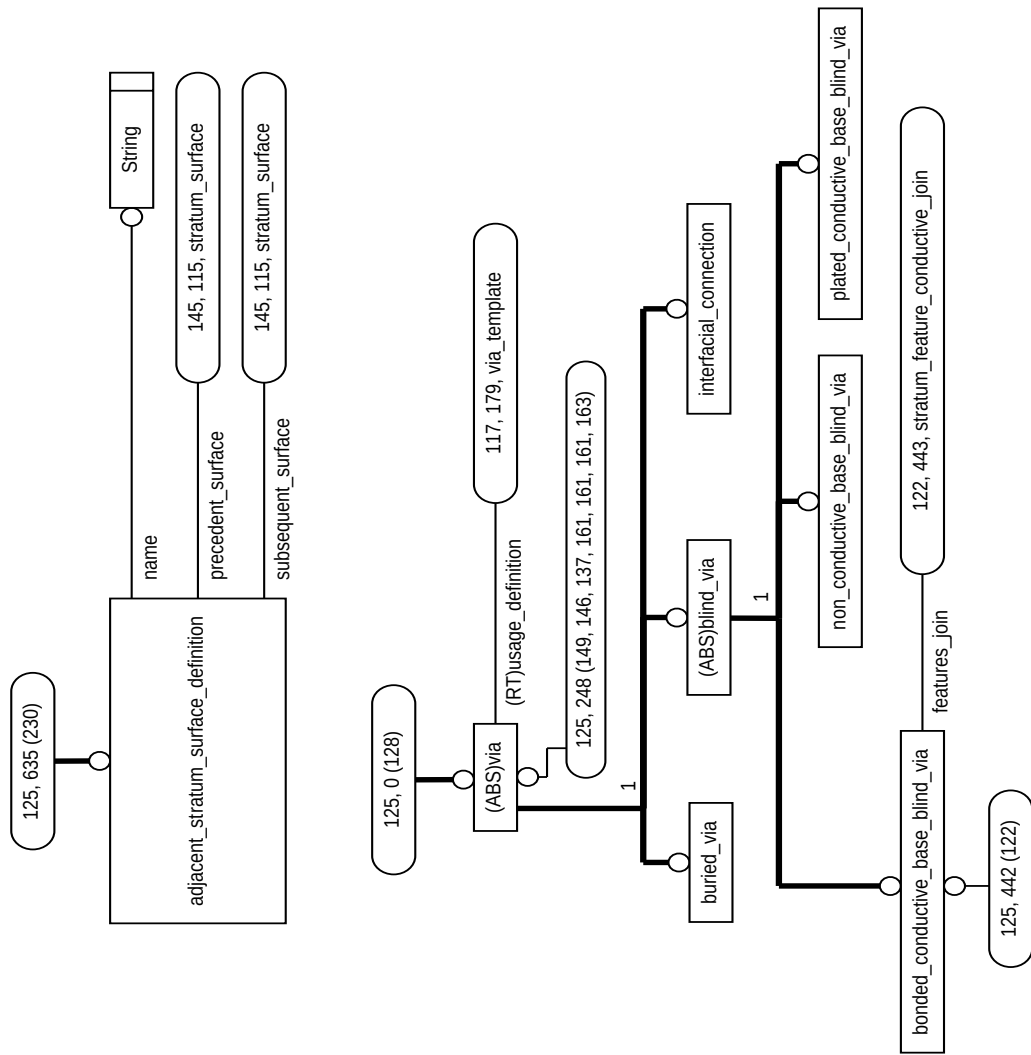


Figure H.125 – ARM EXPRESS-G Graphic 125 of 243

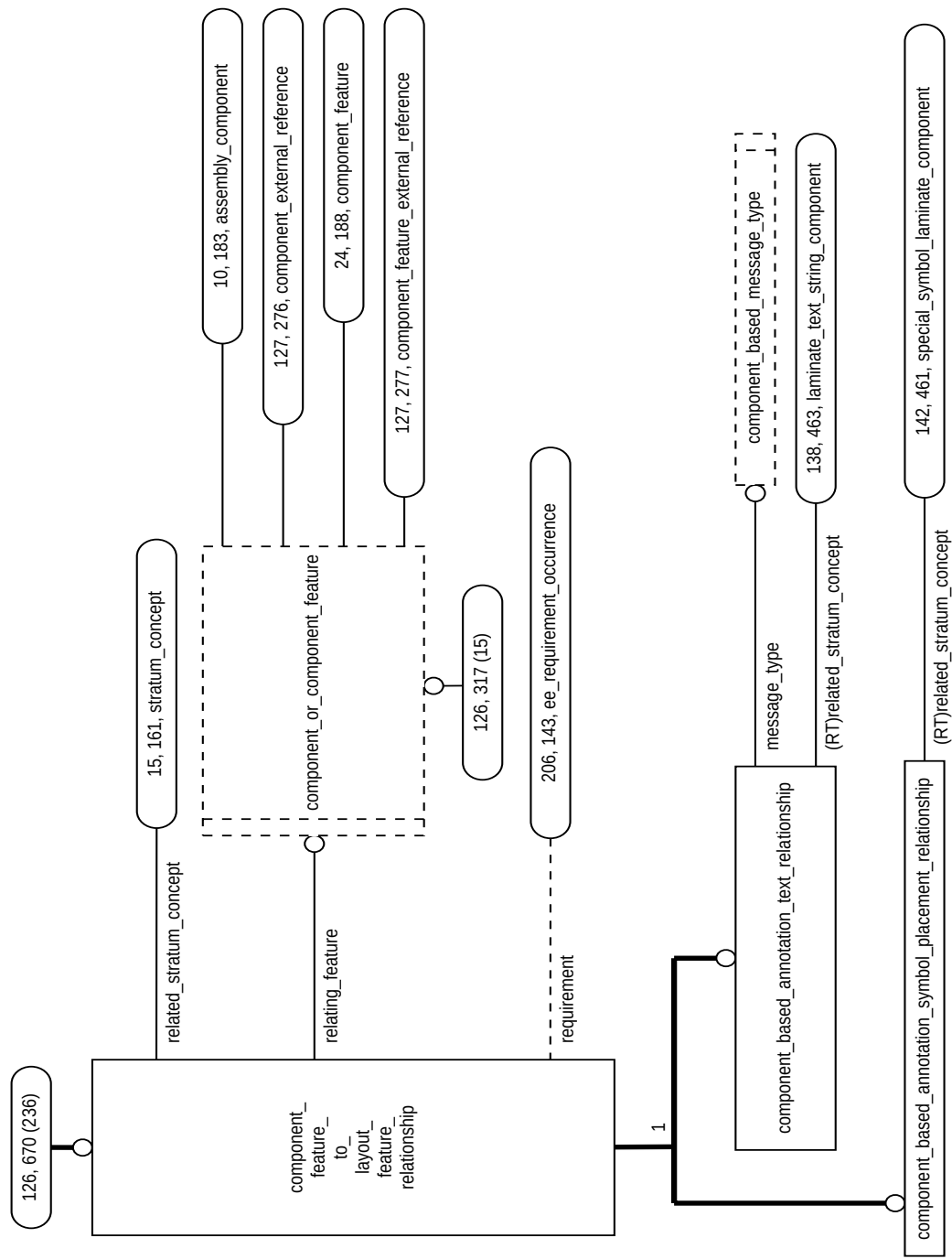


Figure H.126 – ARM EXPRESS-G Graphic 126 of 243

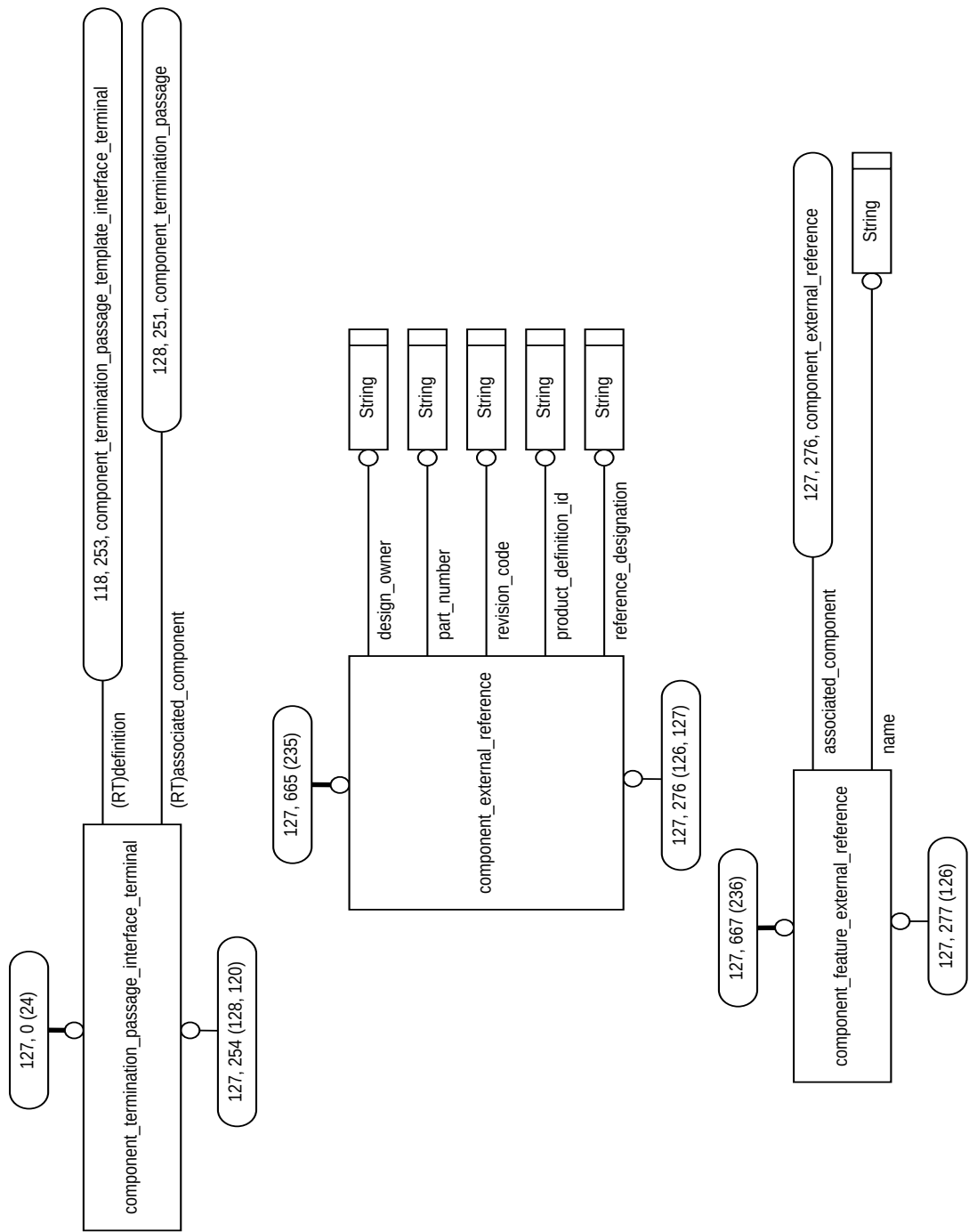


Figure H.127 – ARM EXPRESS-G Graphic 127 of 243

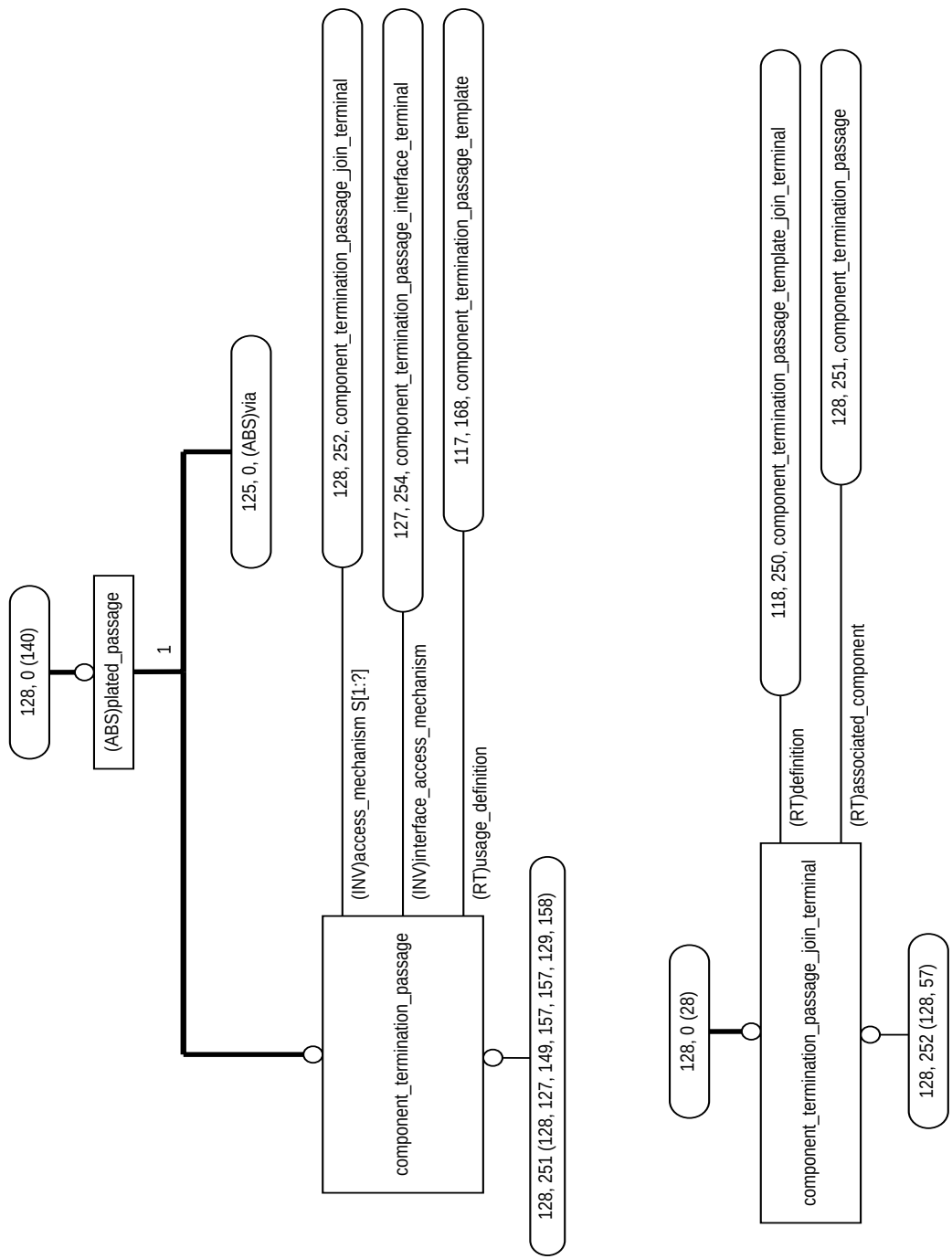


Figure H.128 – ARM EXPRESS-G Graphic 128 of 243

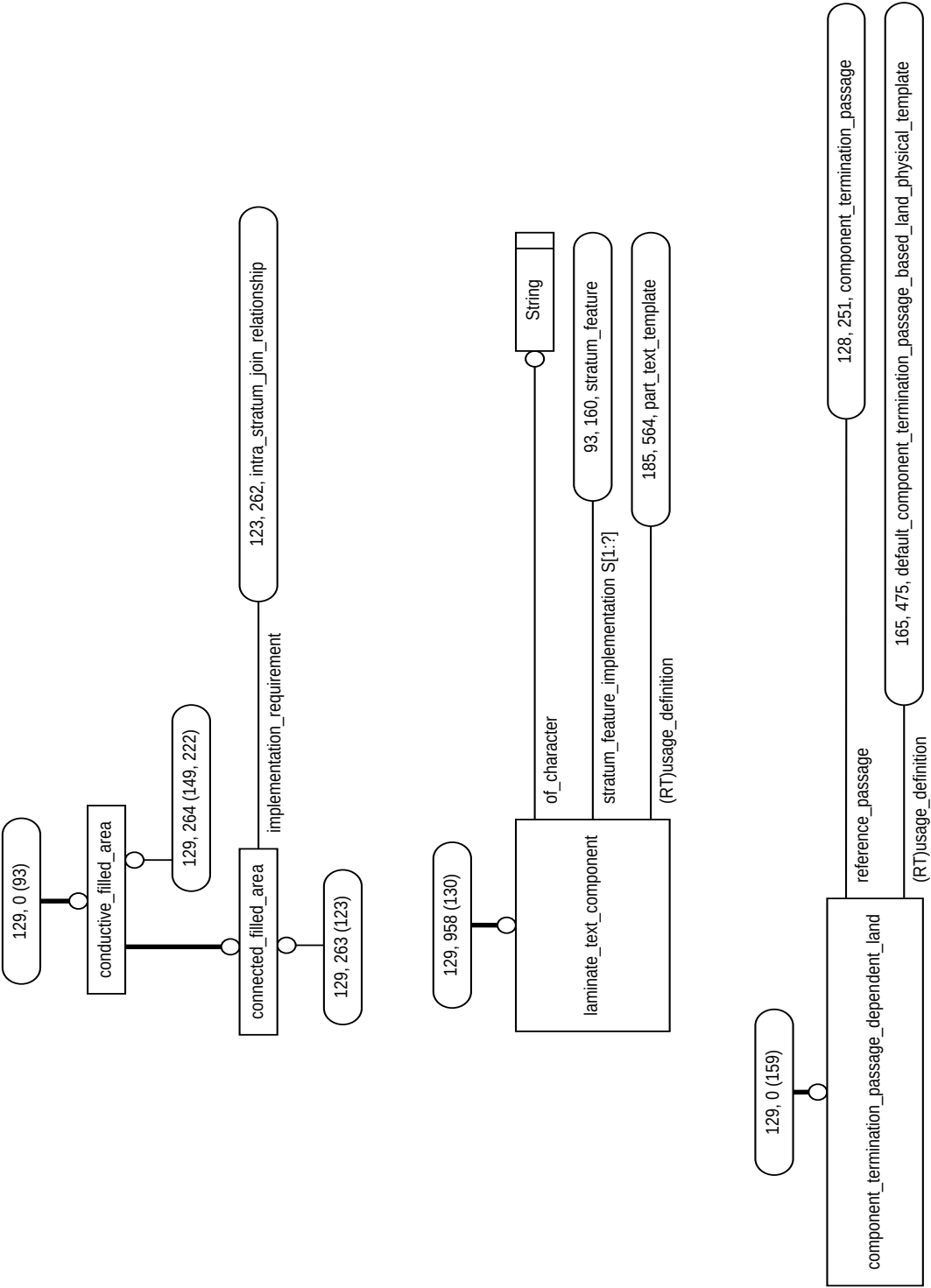


Figure H.129 – ARM EXPRESS-G Graphic 129 of 243

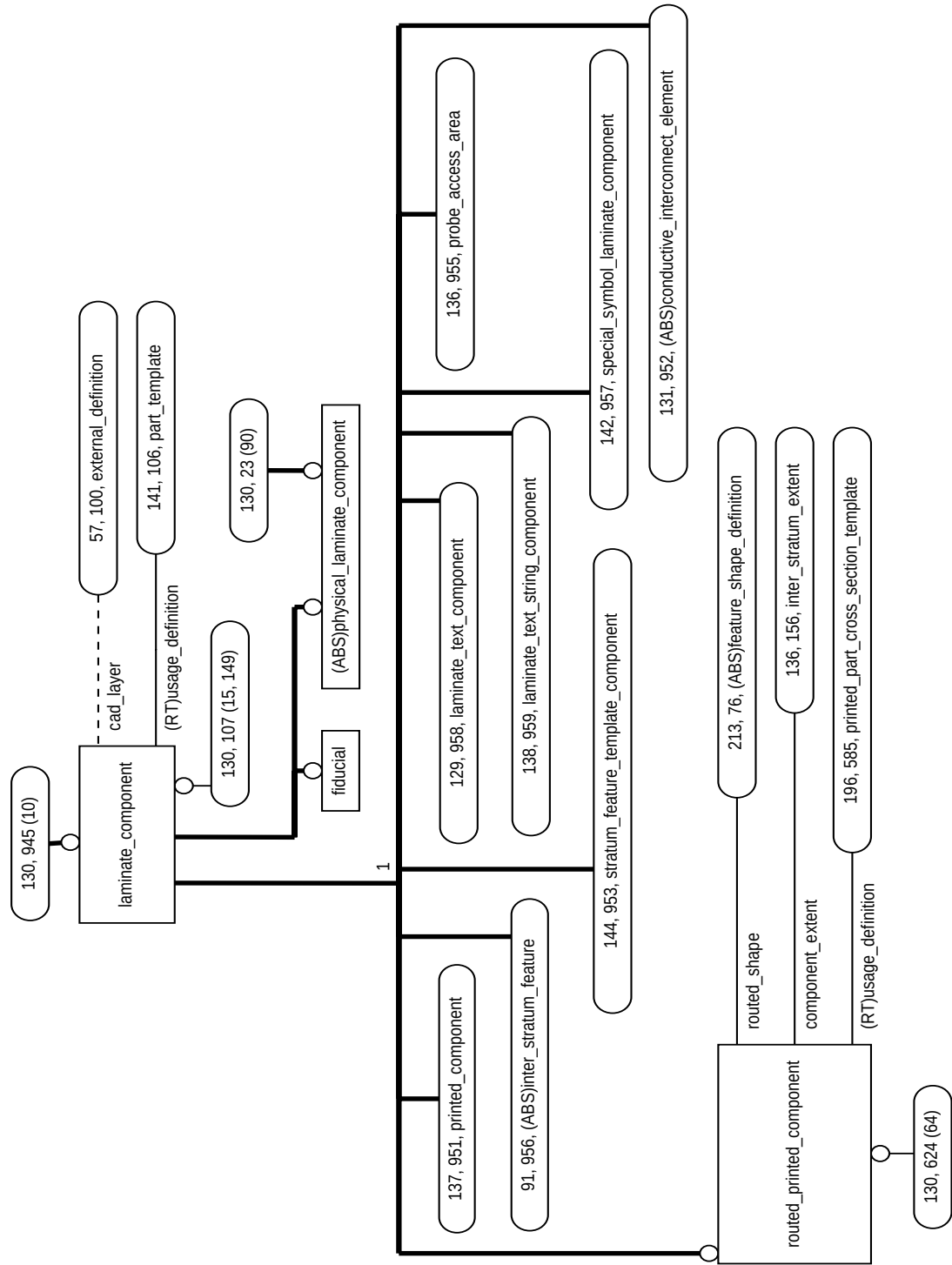


Figure H.130 – ARM EXPRESS-G Graphic 130 of 243

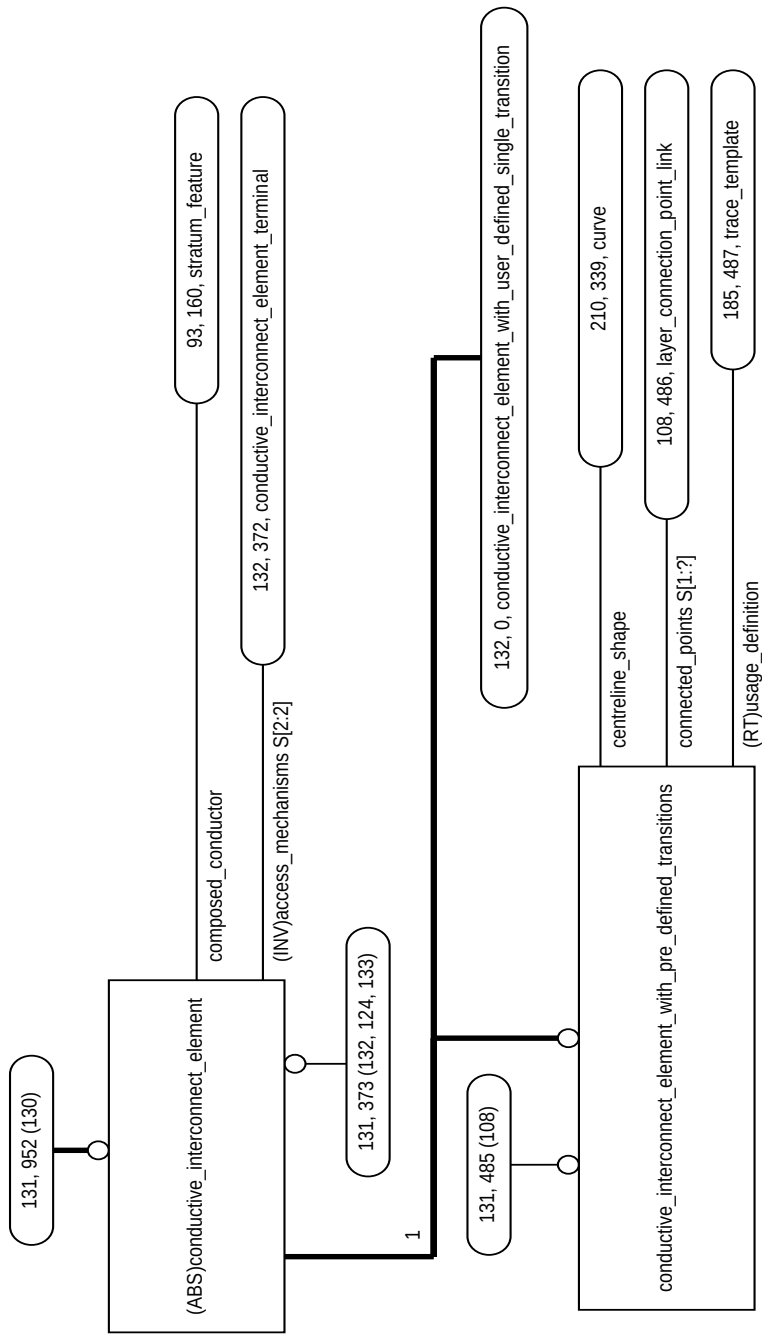


Figure H.131 – ARM EXPRESS-G Graphic 131 of 243

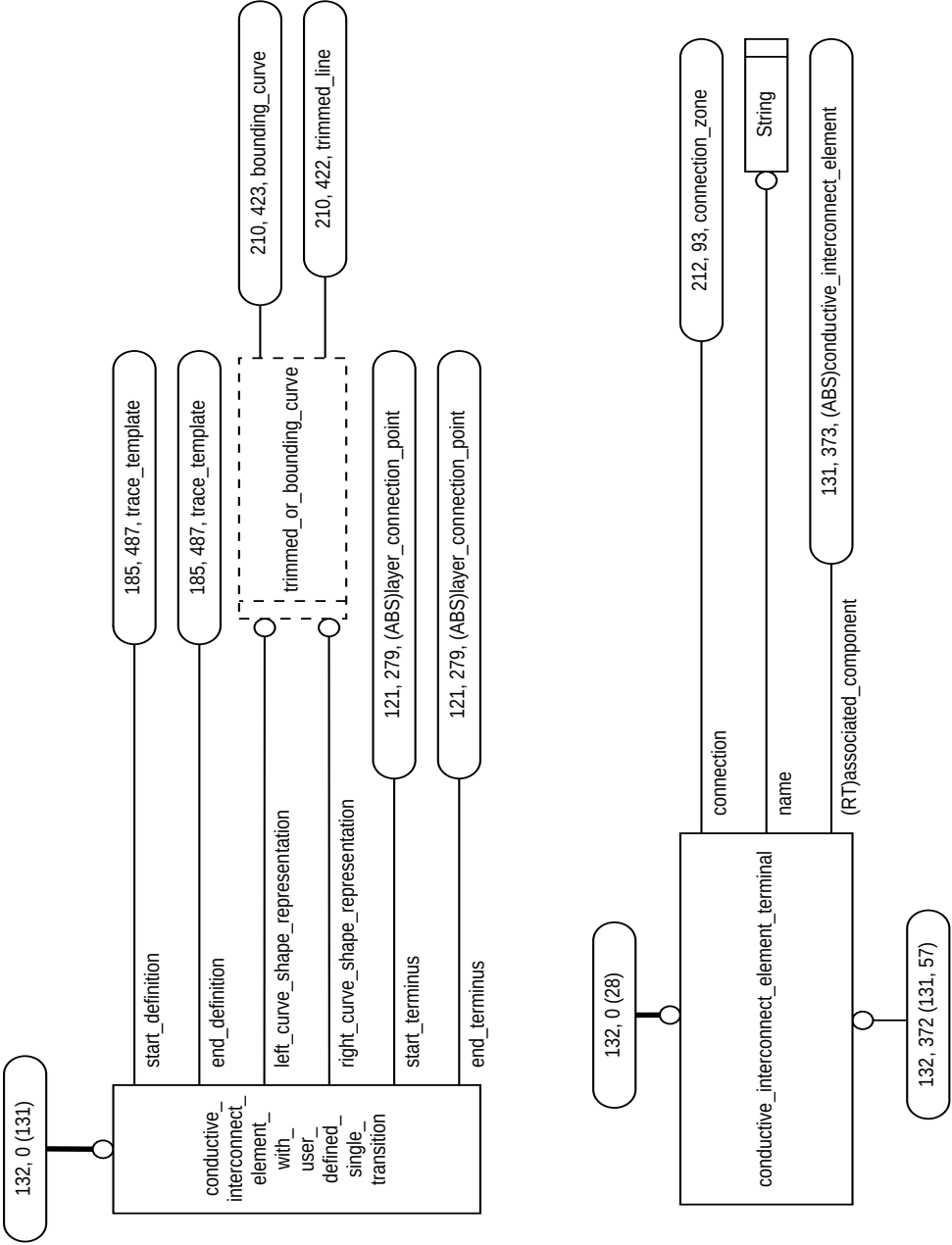


Figure H.132 – ARM EXPRESS-G Graphic 132 of 243

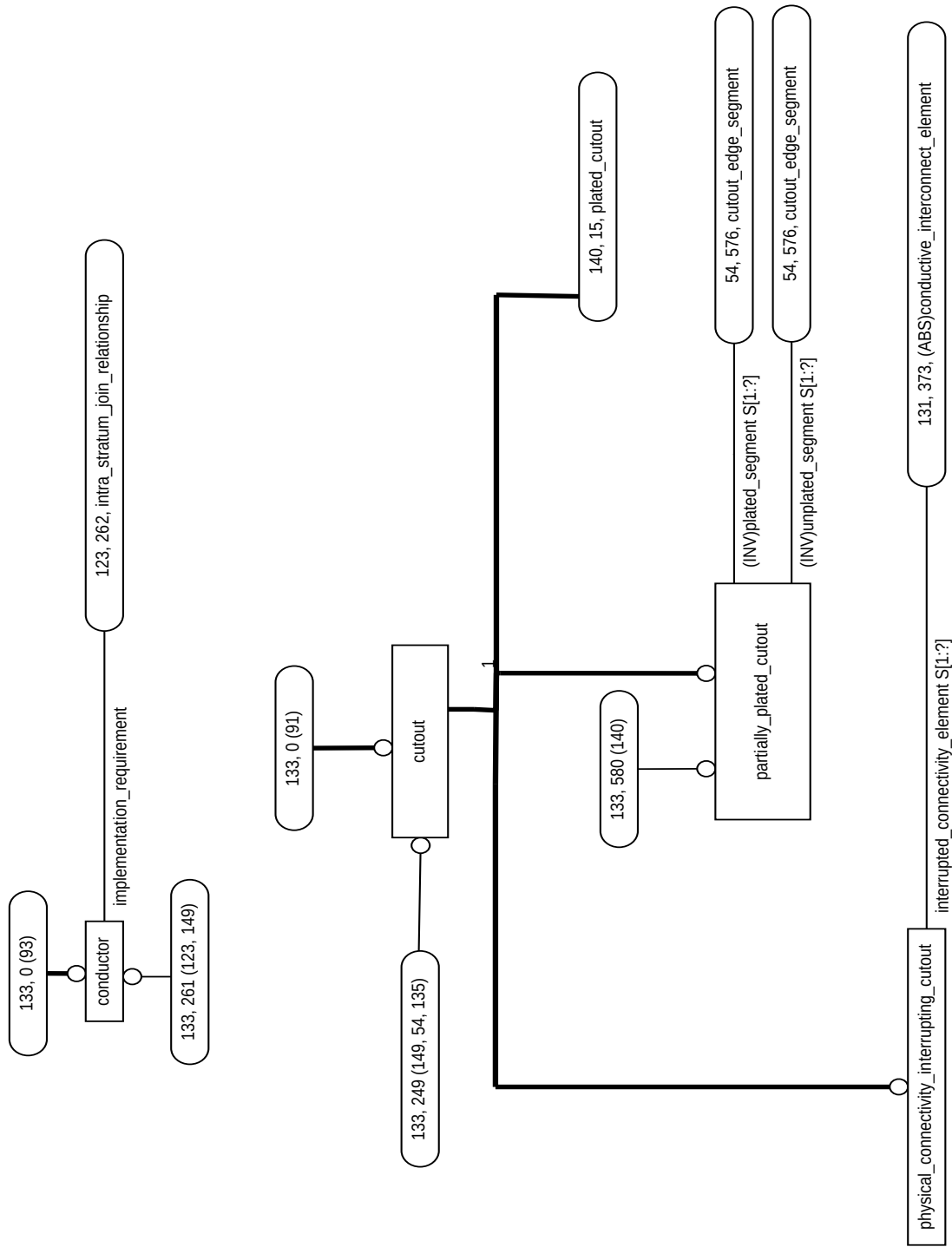


Figure H.133 – ARM EXPRESS-G Graphic 133 of 243

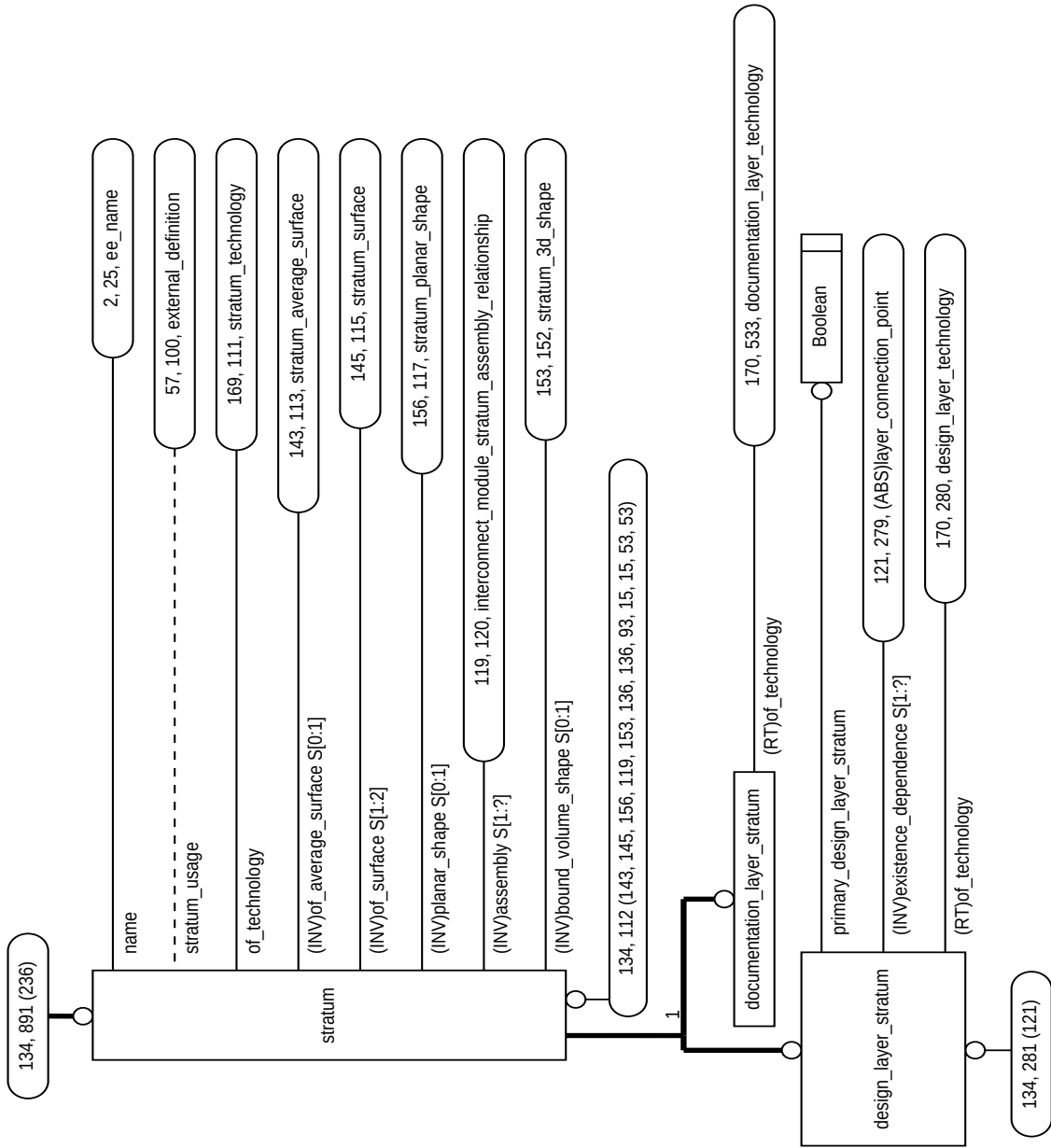


Figure H.134 – ARM EXPRESS-G Graphic 134 of 243

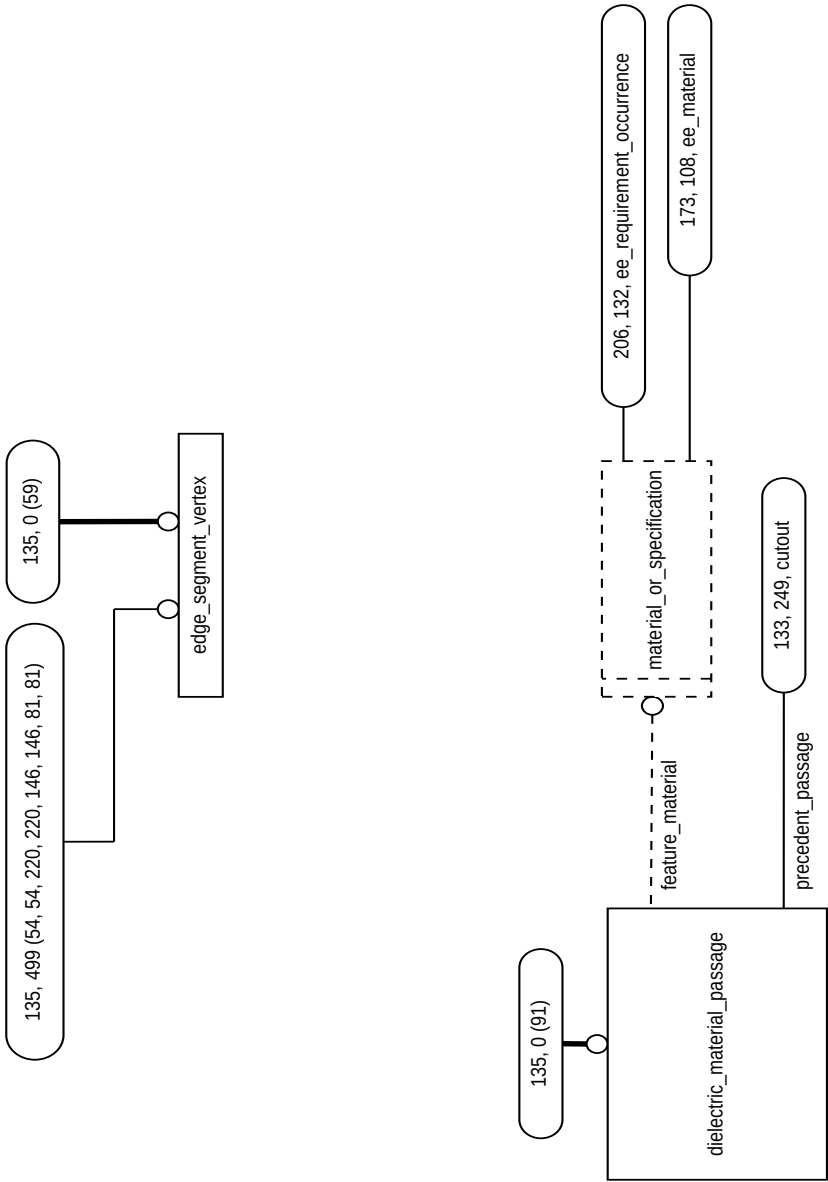


Figure H.135 – ARM EXPRESS-G Graphic 135 of 243

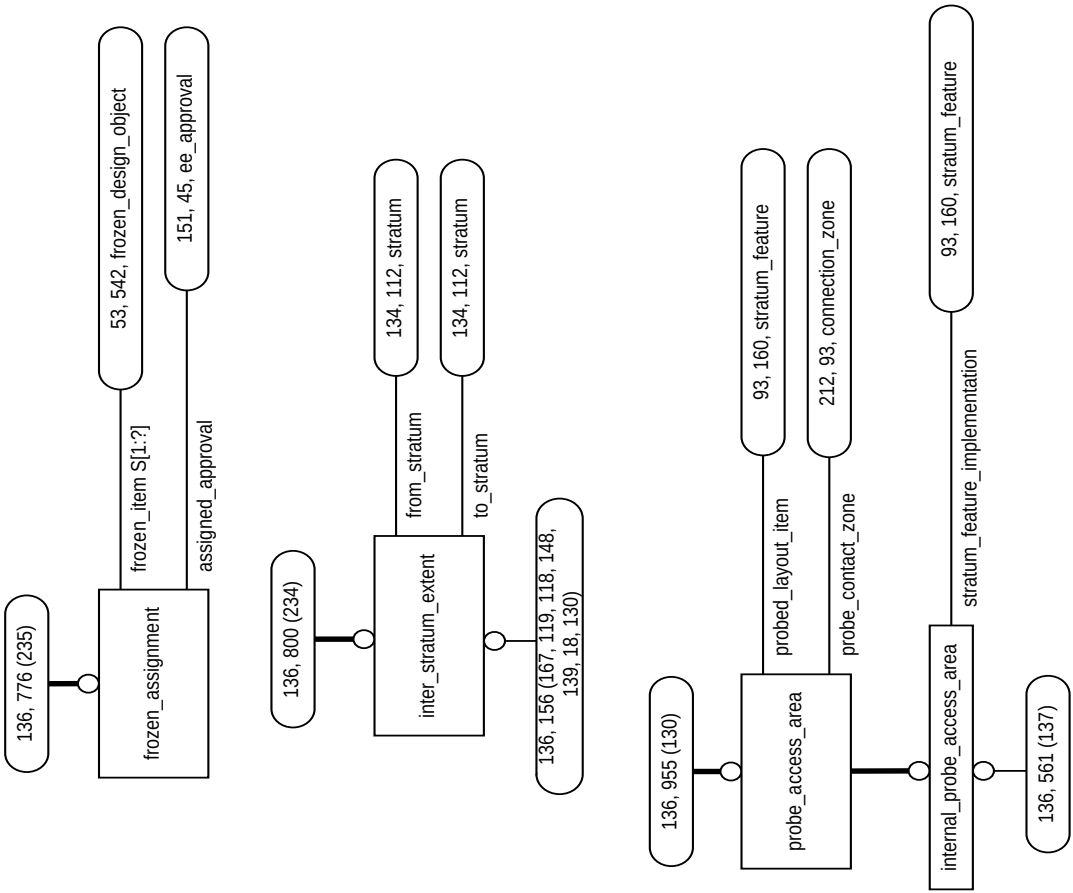


Figure H.136 – ARM EXPRESS-G Graphic 136 of 243

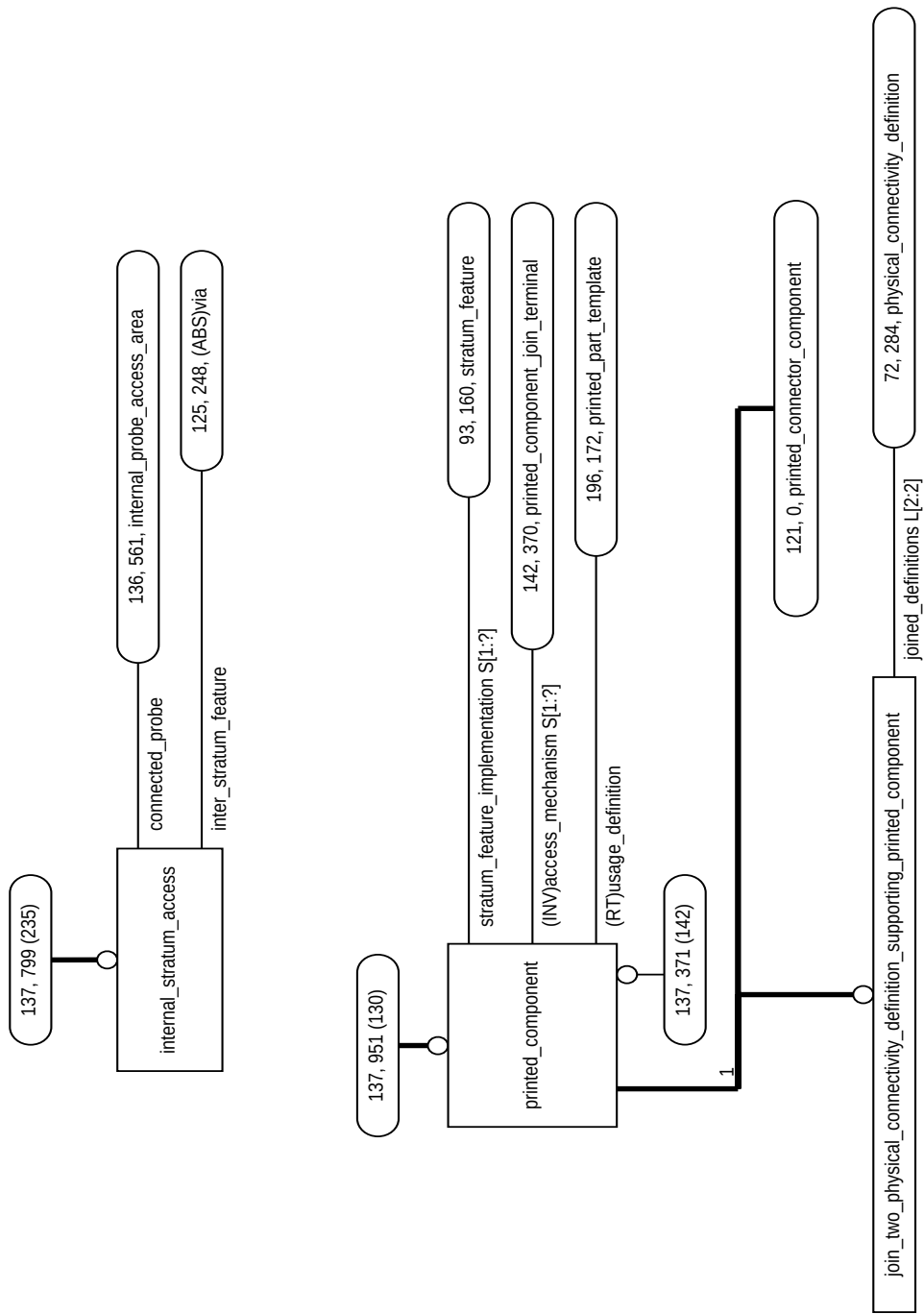


Figure H.137 – ARM EXPRESS-G Graphic 137 of 243

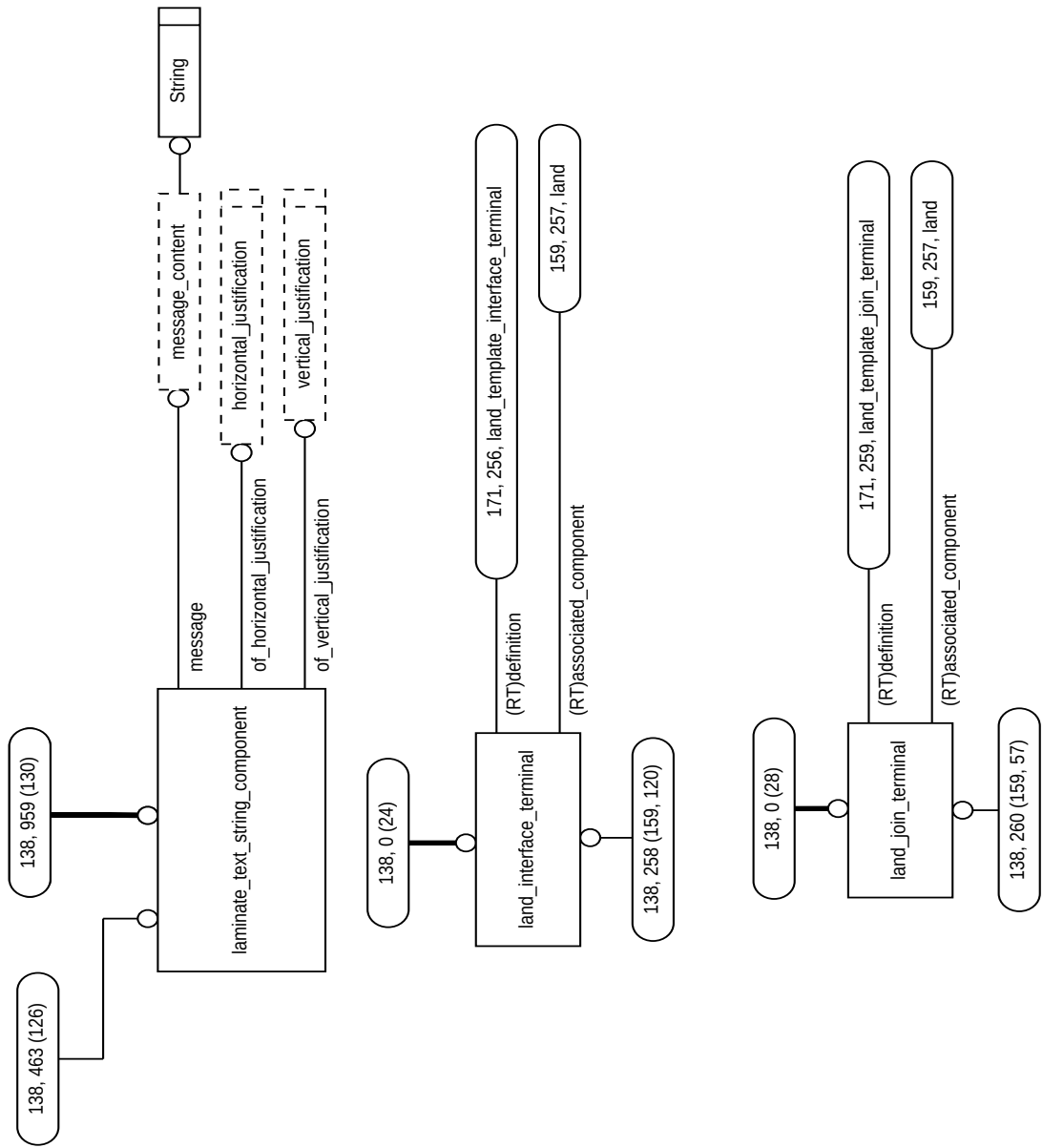


Figure H.138 – ARM EXPRESS-G Graphic 138 of 243

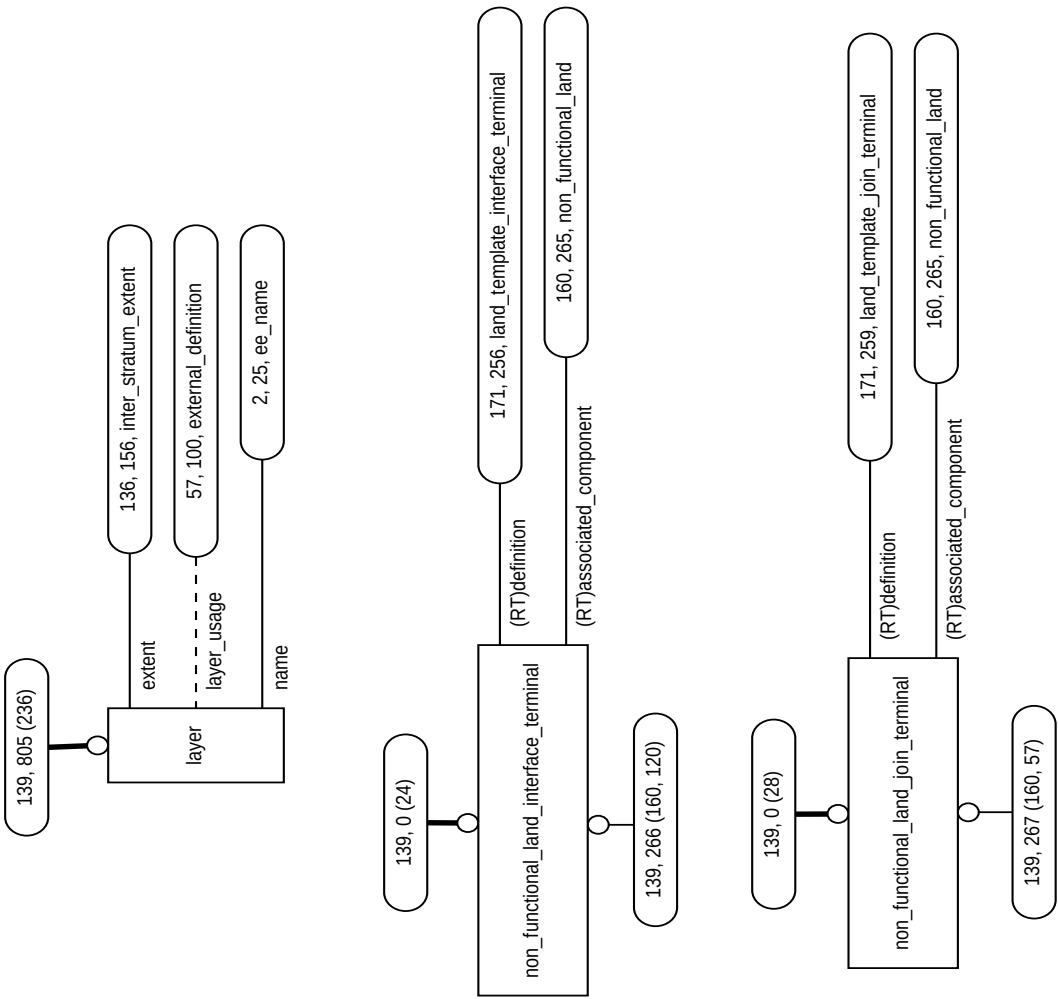
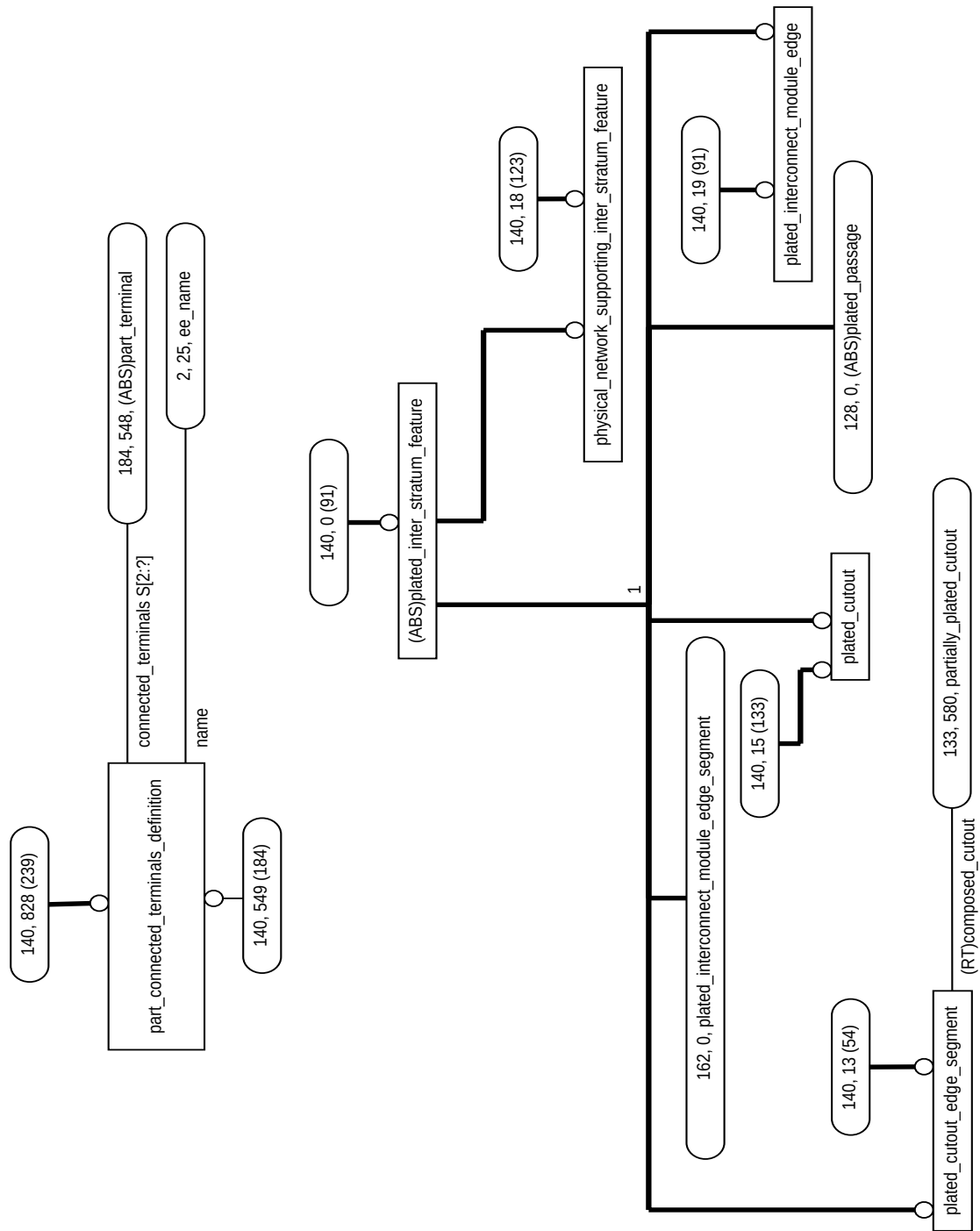


Figure H.139 – ARM EXPRESS-G Graphic 139 of 243



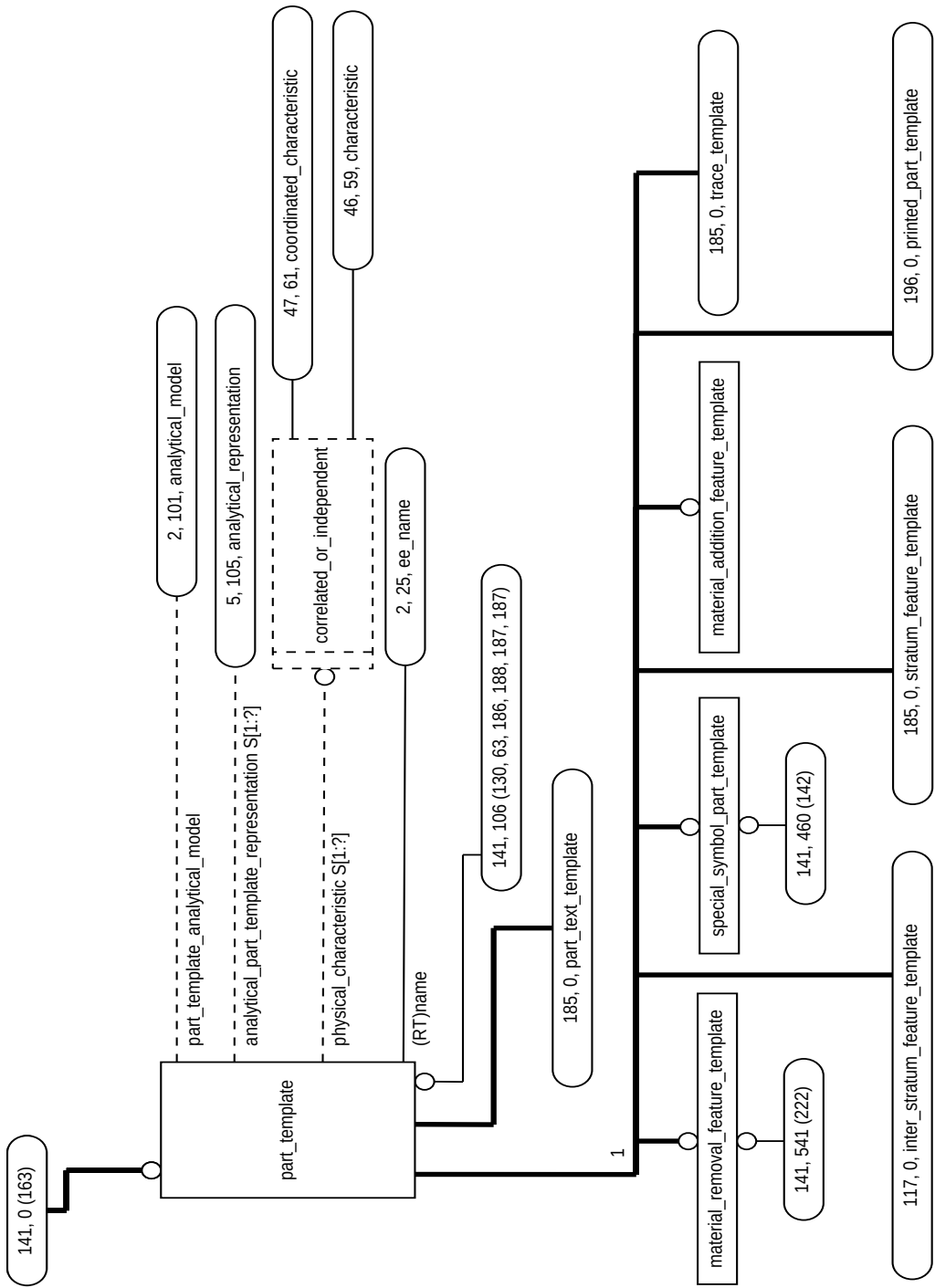


Figure H.141 – ARM EXPRESS-G Graphic 141 of 243

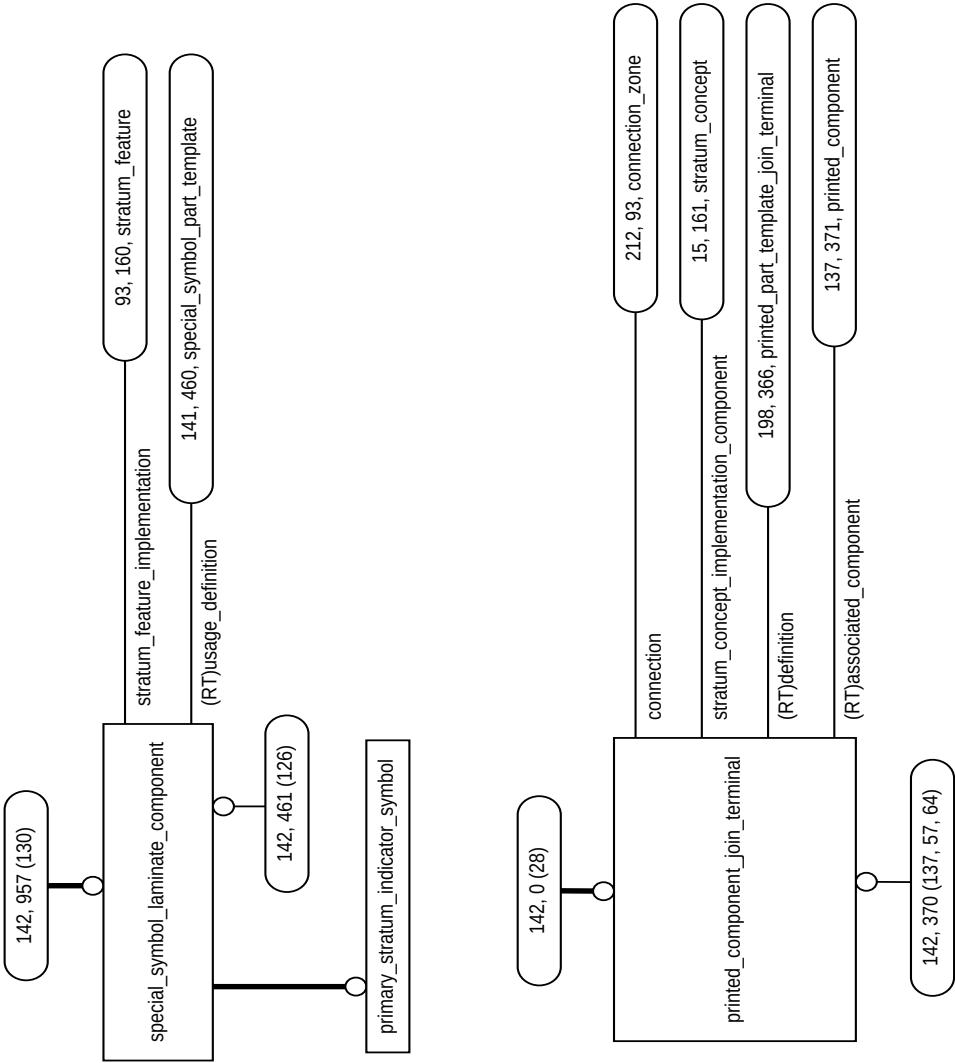


Figure H.142 – ARM EXPRESS-G Graphic 142 of 243

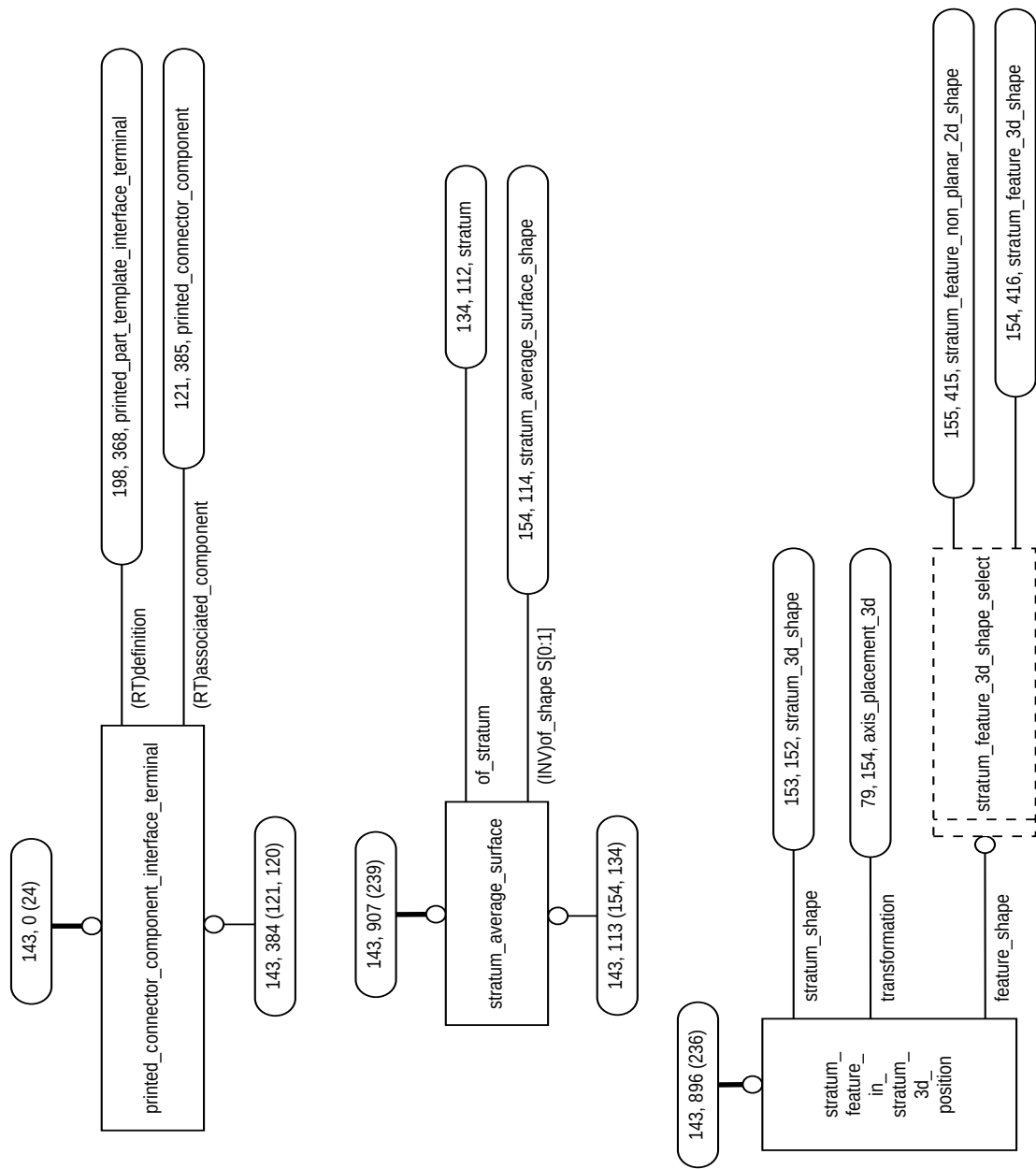


Figure H.143 – ARM EXPRESS-G Graphic 143 of 243

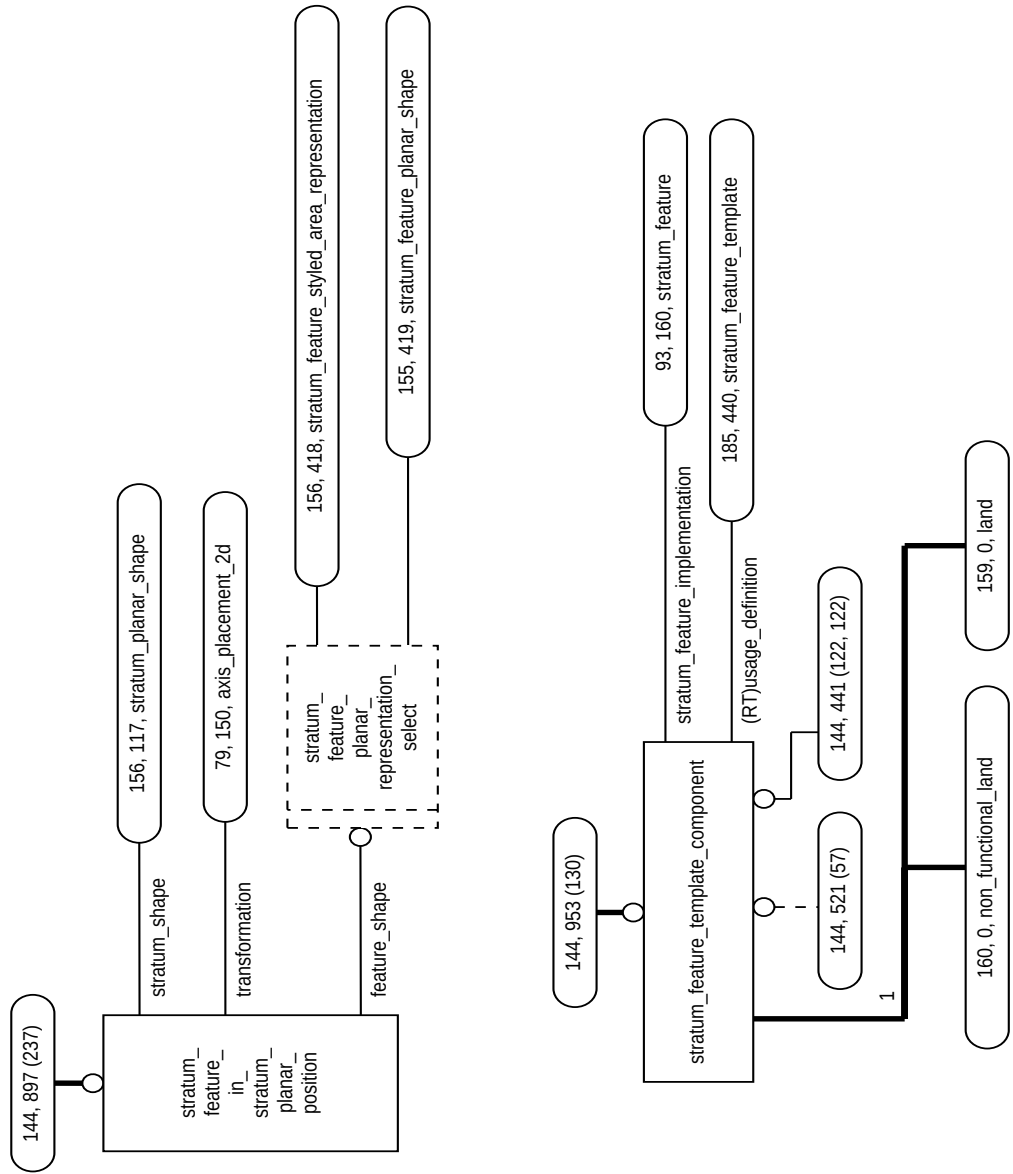


Figure H.144 – ARM EXPRESS-G Graphic 144 of 243

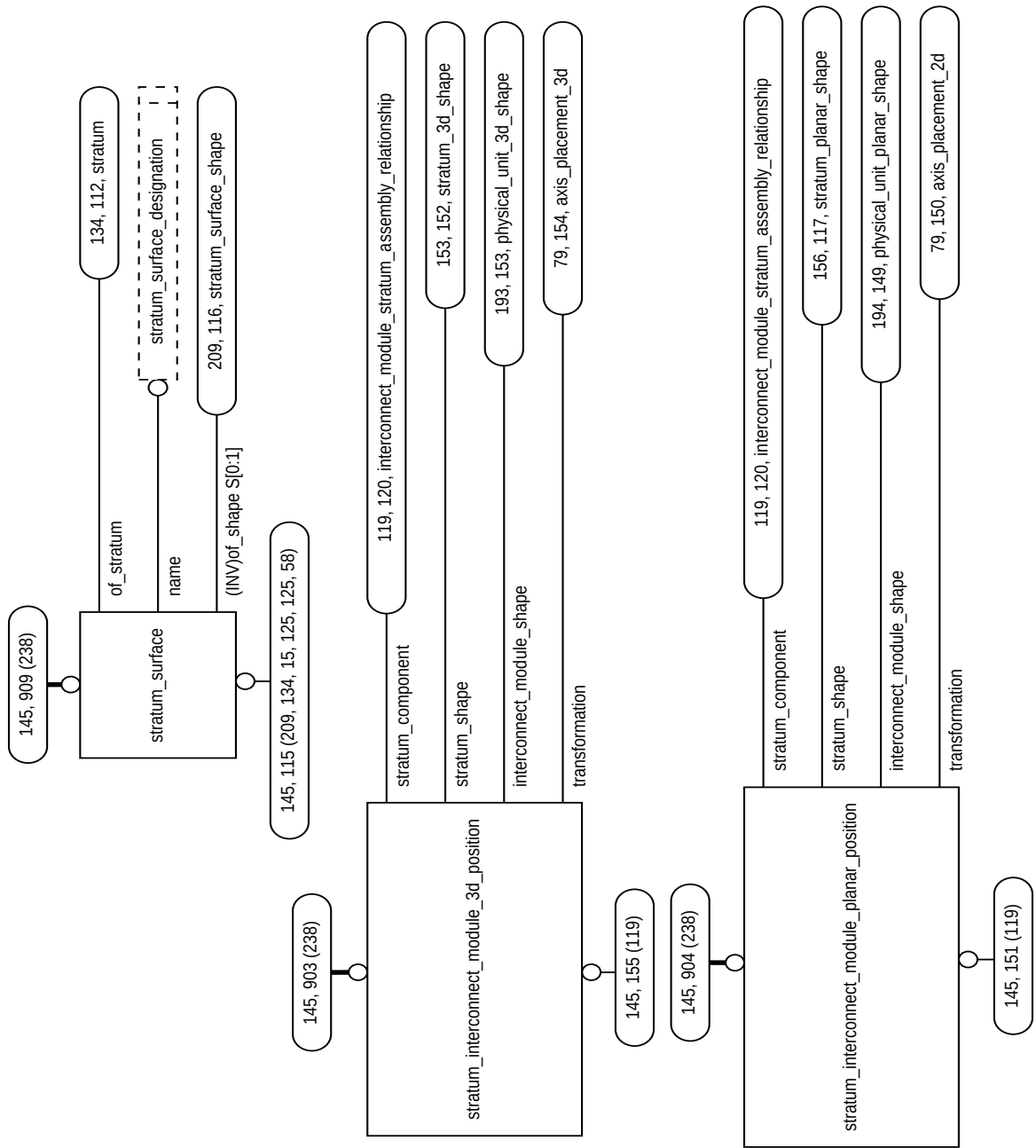


Figure H.145 – ARM EXPRESS-G Graphic 145 of 243

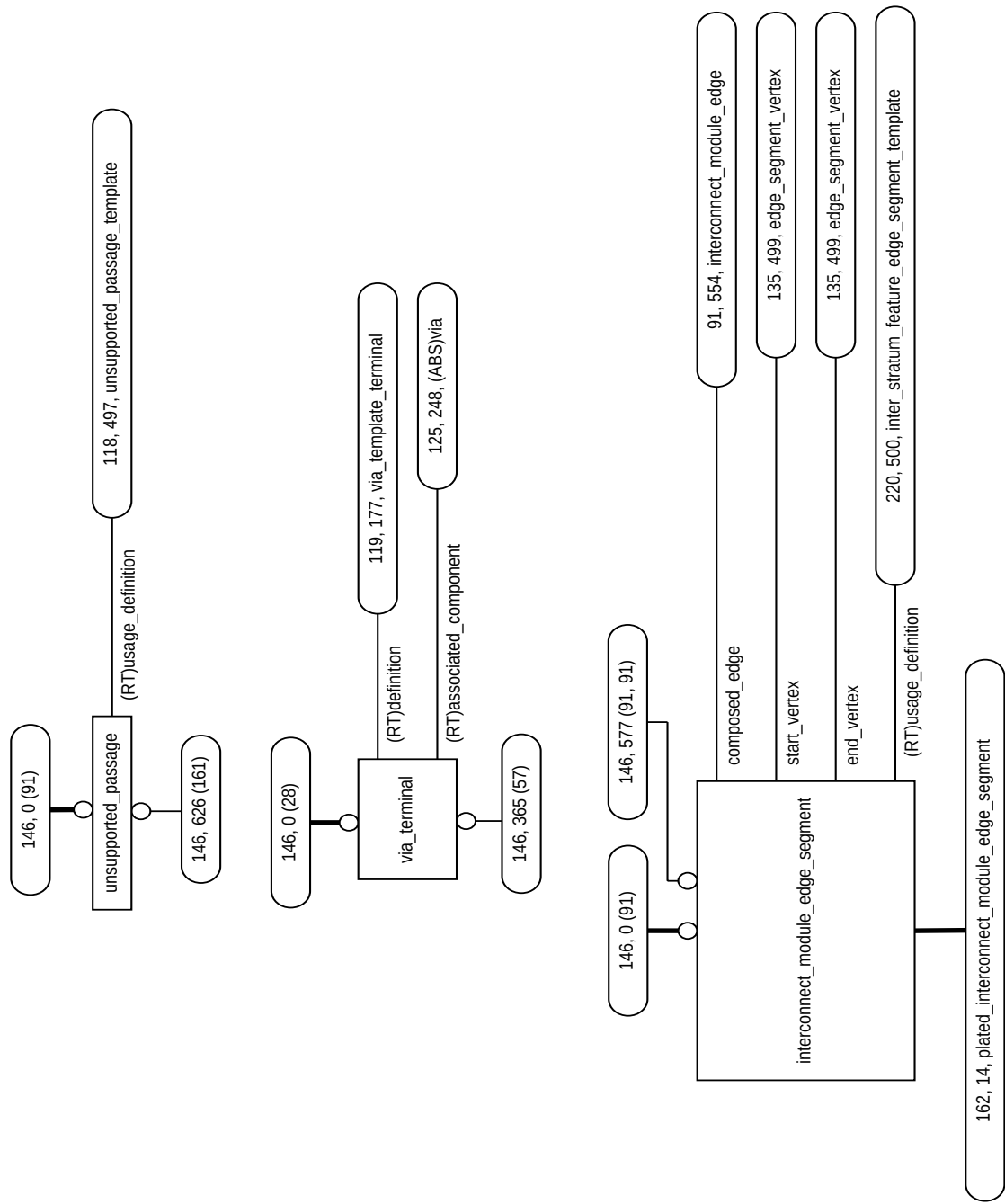


Figure H.146 – ARM EXPRESS-G Graphic 146 of 243

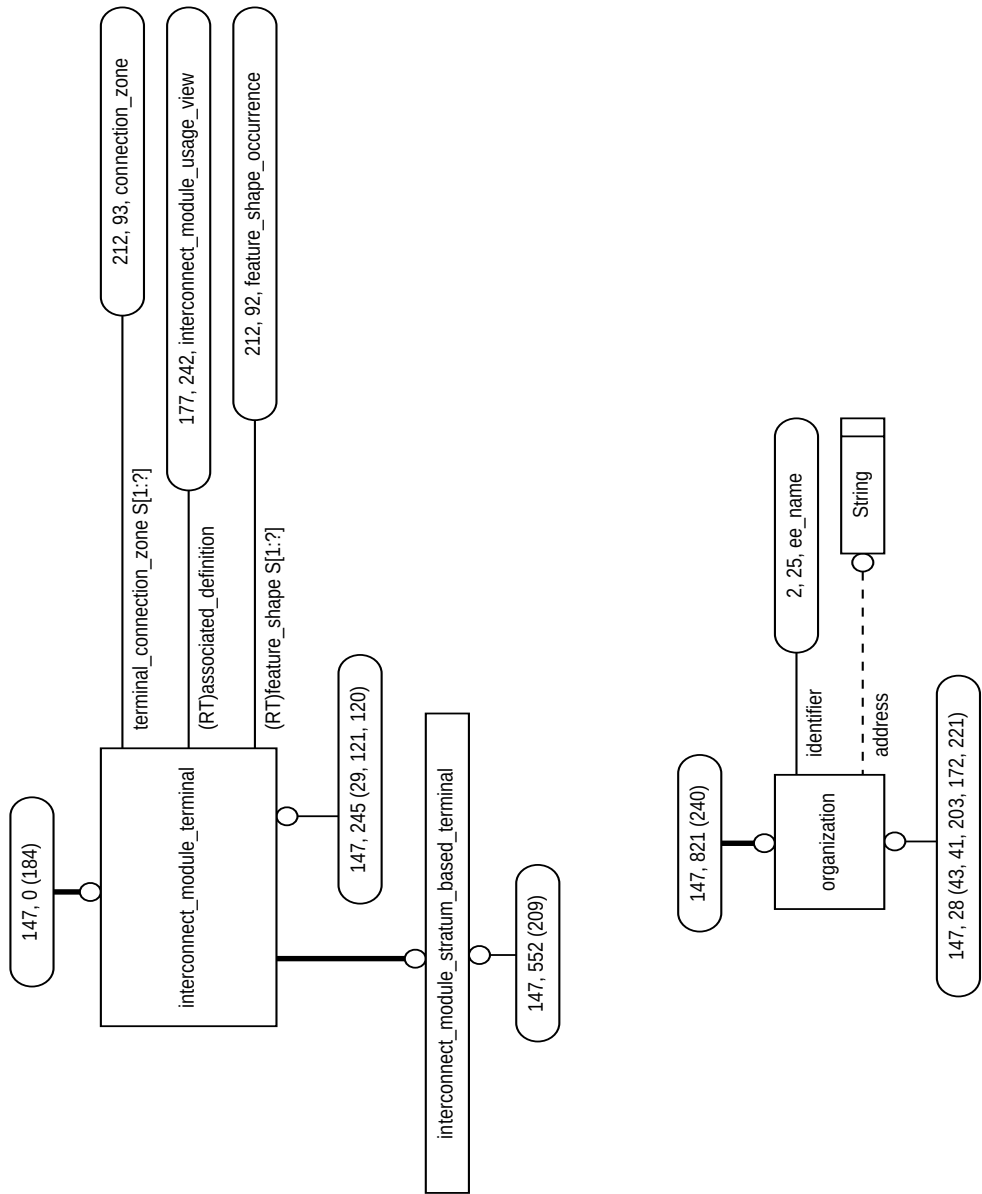


Figure H.147 – ARM EXPRESS-G Graphic 147 of 243

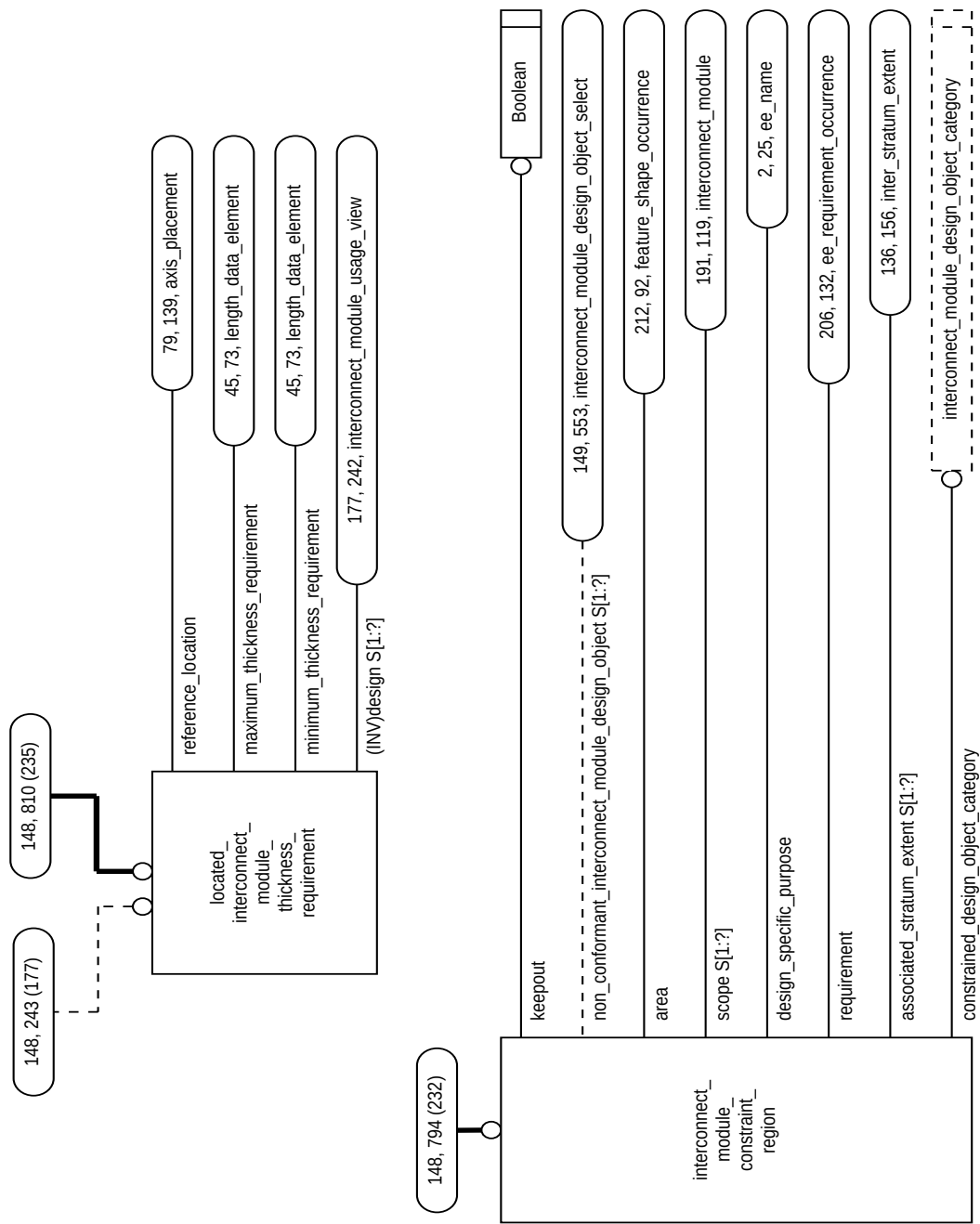


Figure H.148 – ARM EXPRESS-G Graphic 148 of 243

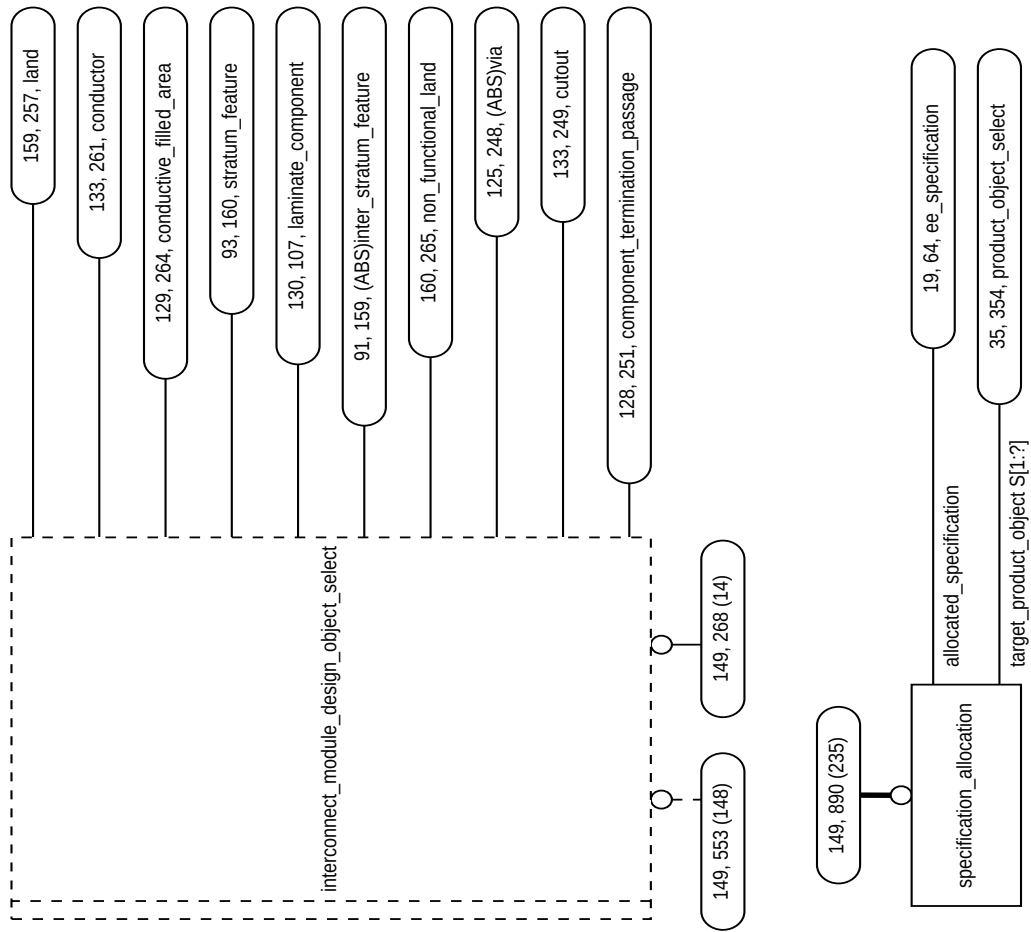


Figure H.149 – ARM EXPRESS-G Graphic 149 of 243

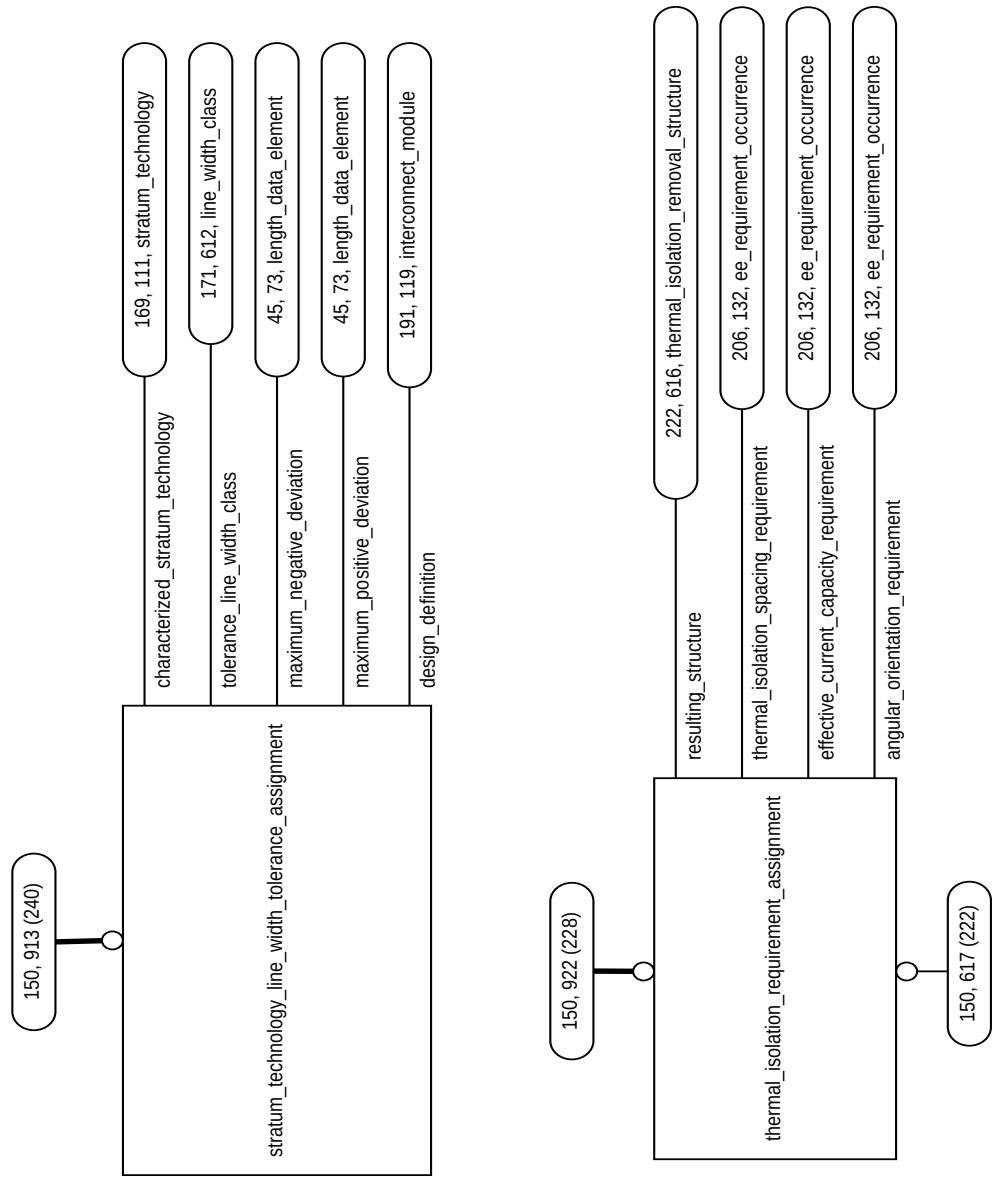


Figure H.150 – ARM EXPRESS-G Graphic 150 of 243

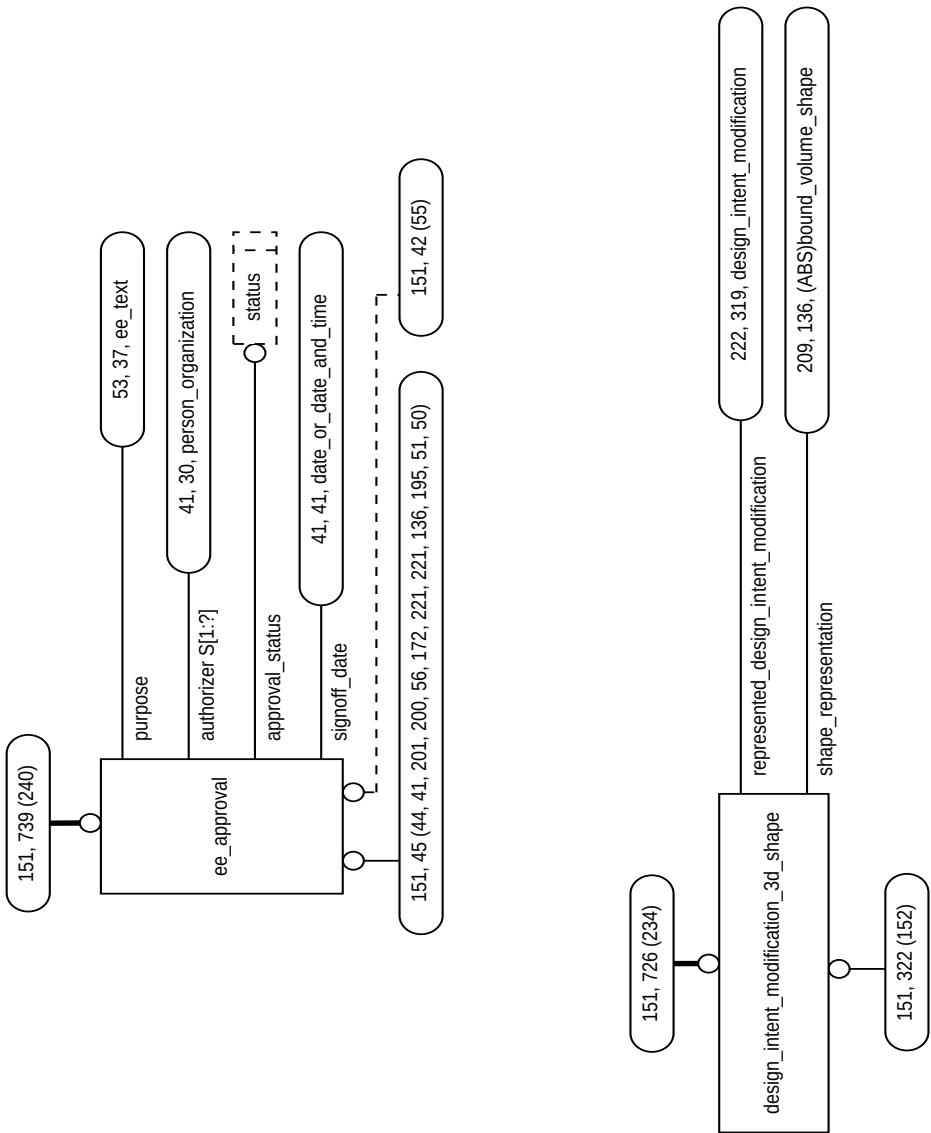


Figure H.151 – ARM EXPRESS-G Graphic 151 of 243

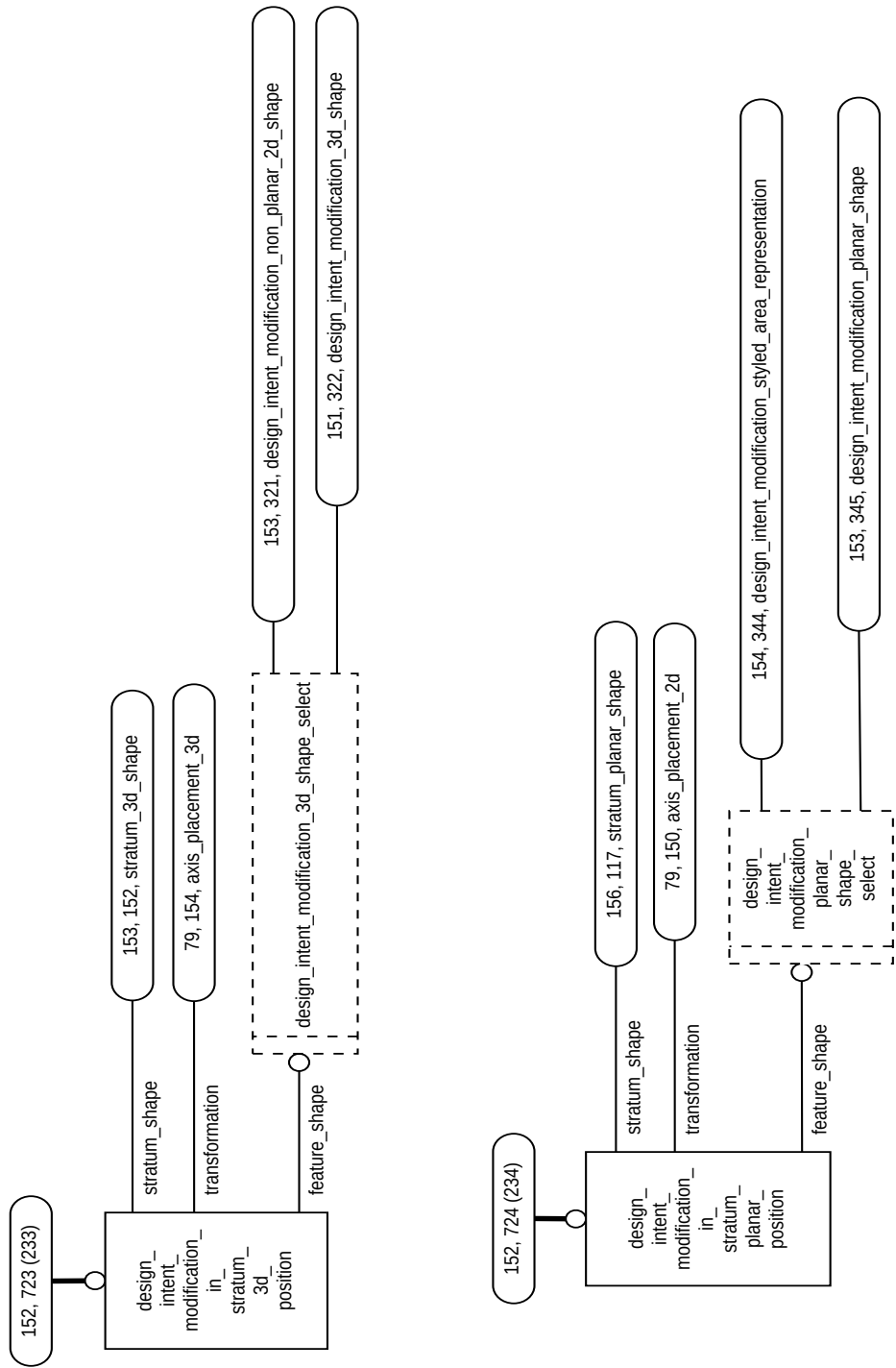


Figure H.152 – ARM EXPRESS-G Graphic 152 of 243

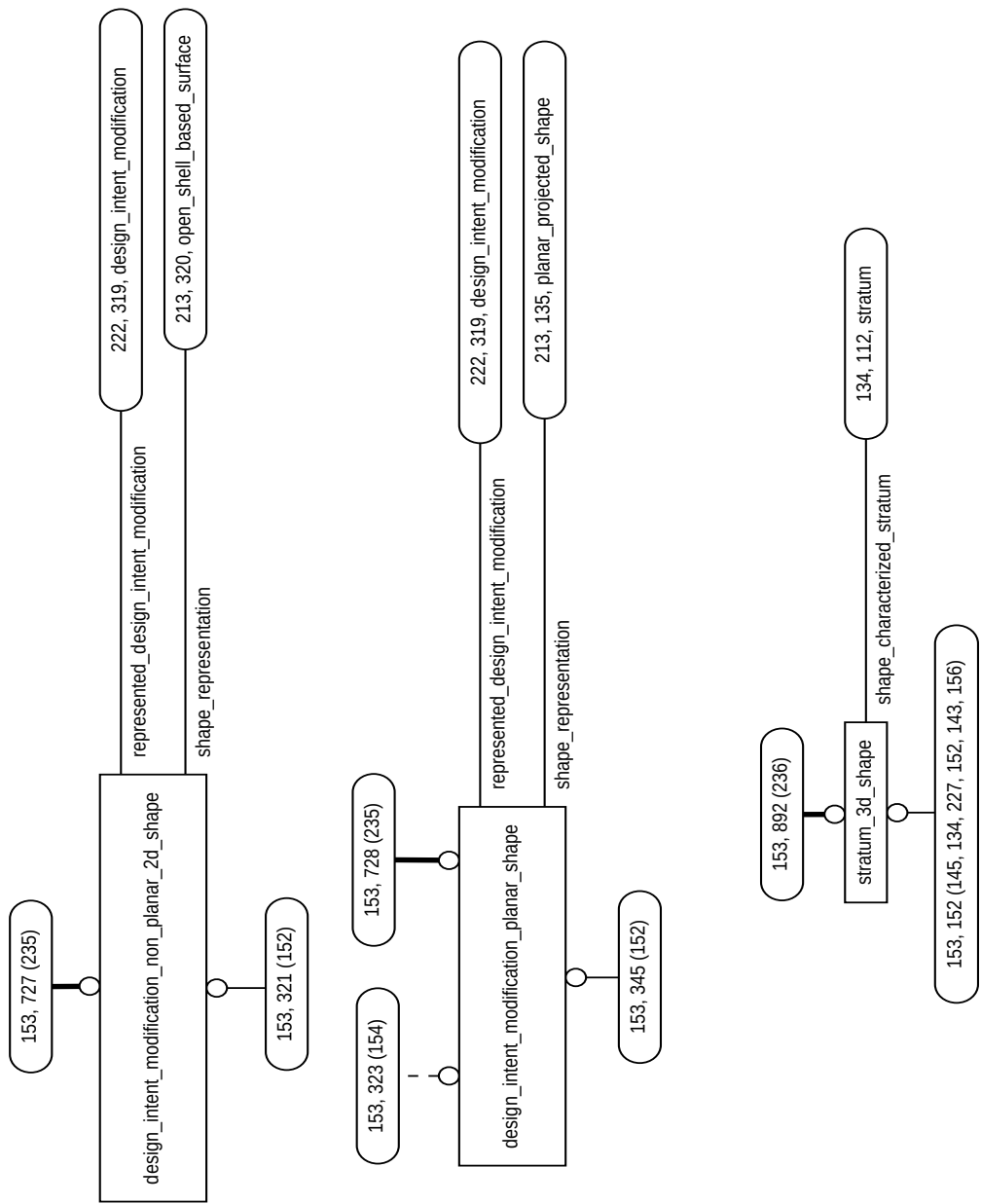


Figure H.153 – ARM EXPRESS-G Graphic 153 of 243

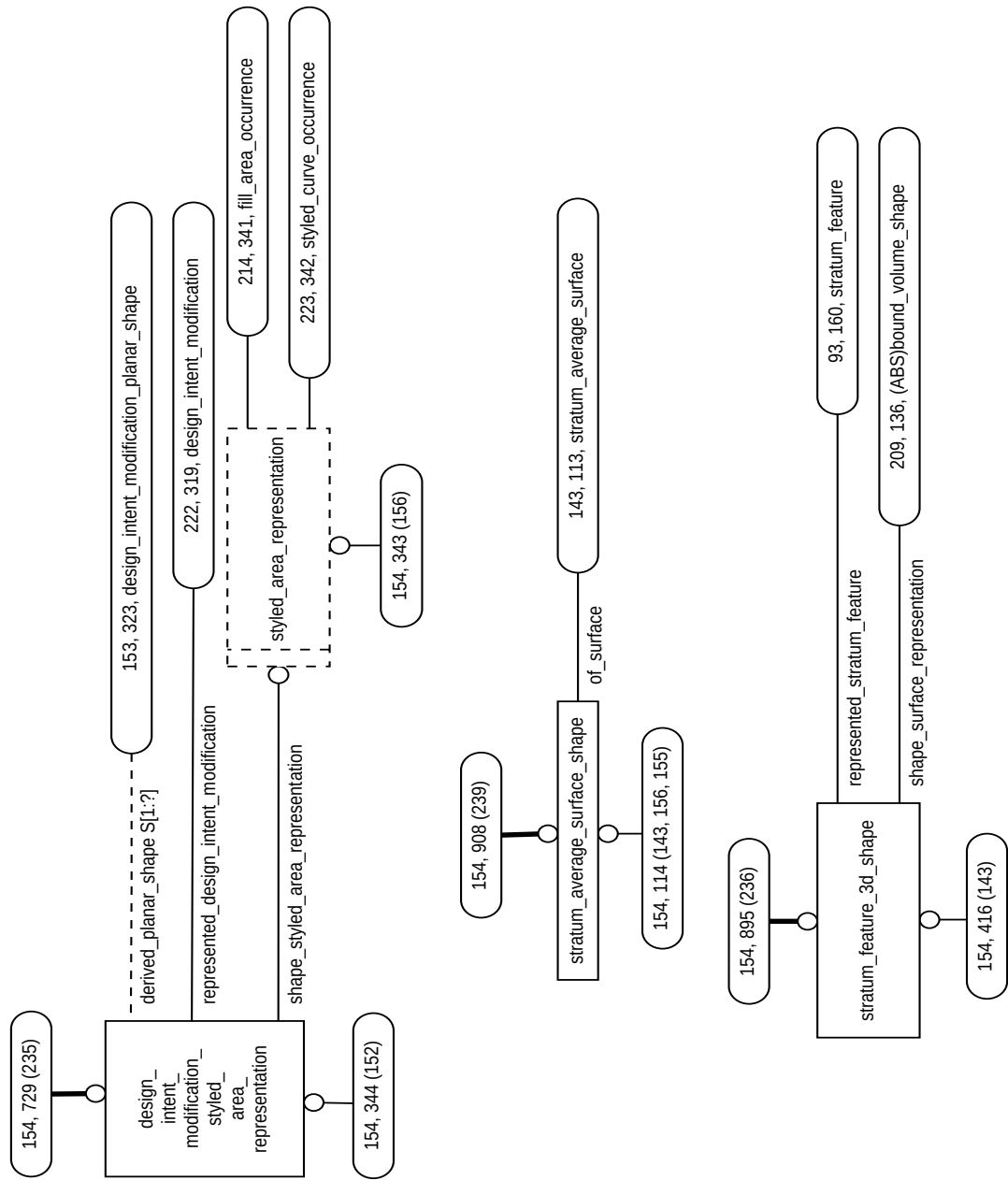


Figure H.154 – ARM EXPRESS-G Graphic 154 of 243

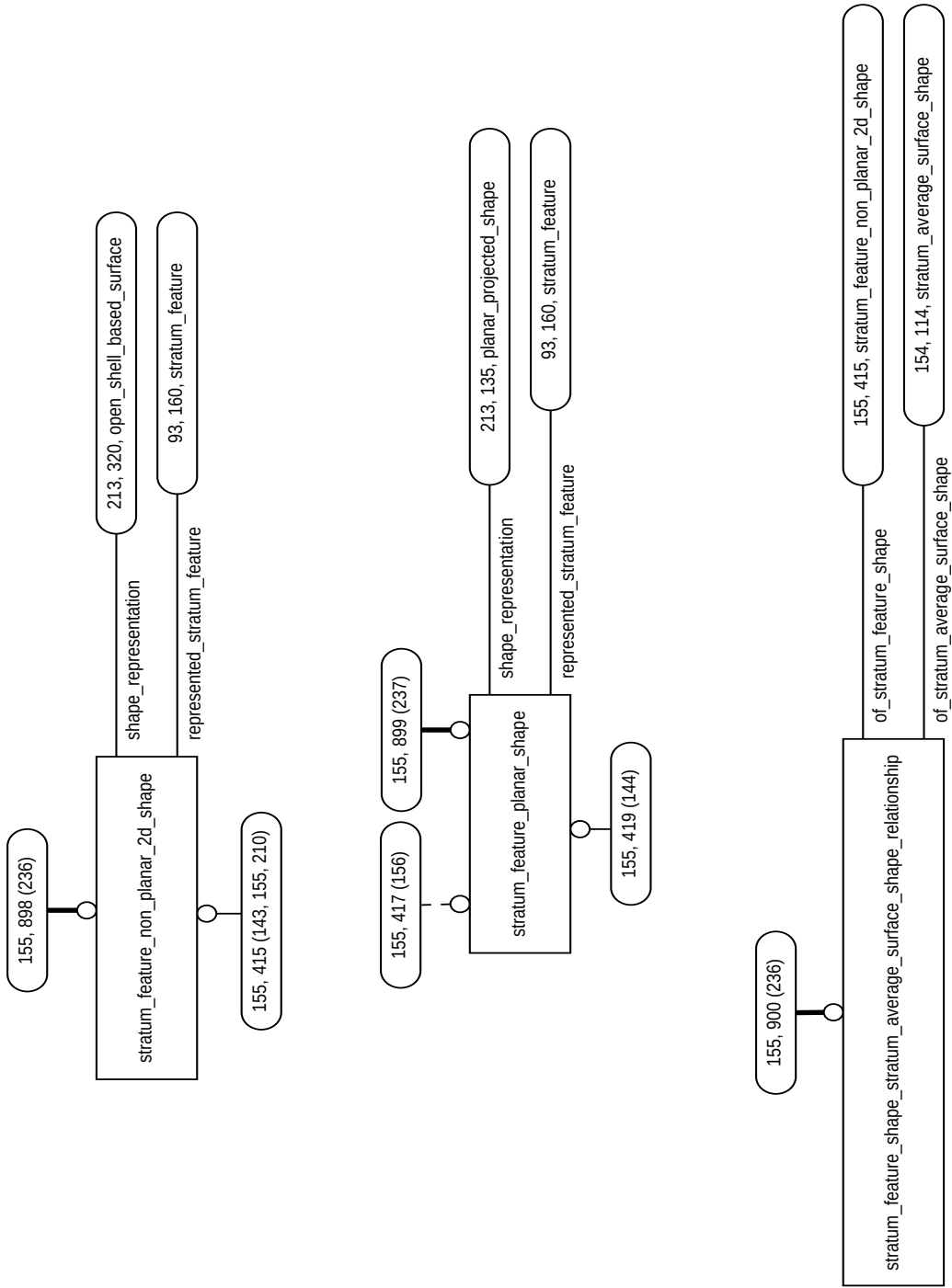


Figure H.155 – ARM EXPRESS-G Graphic 155 of 243

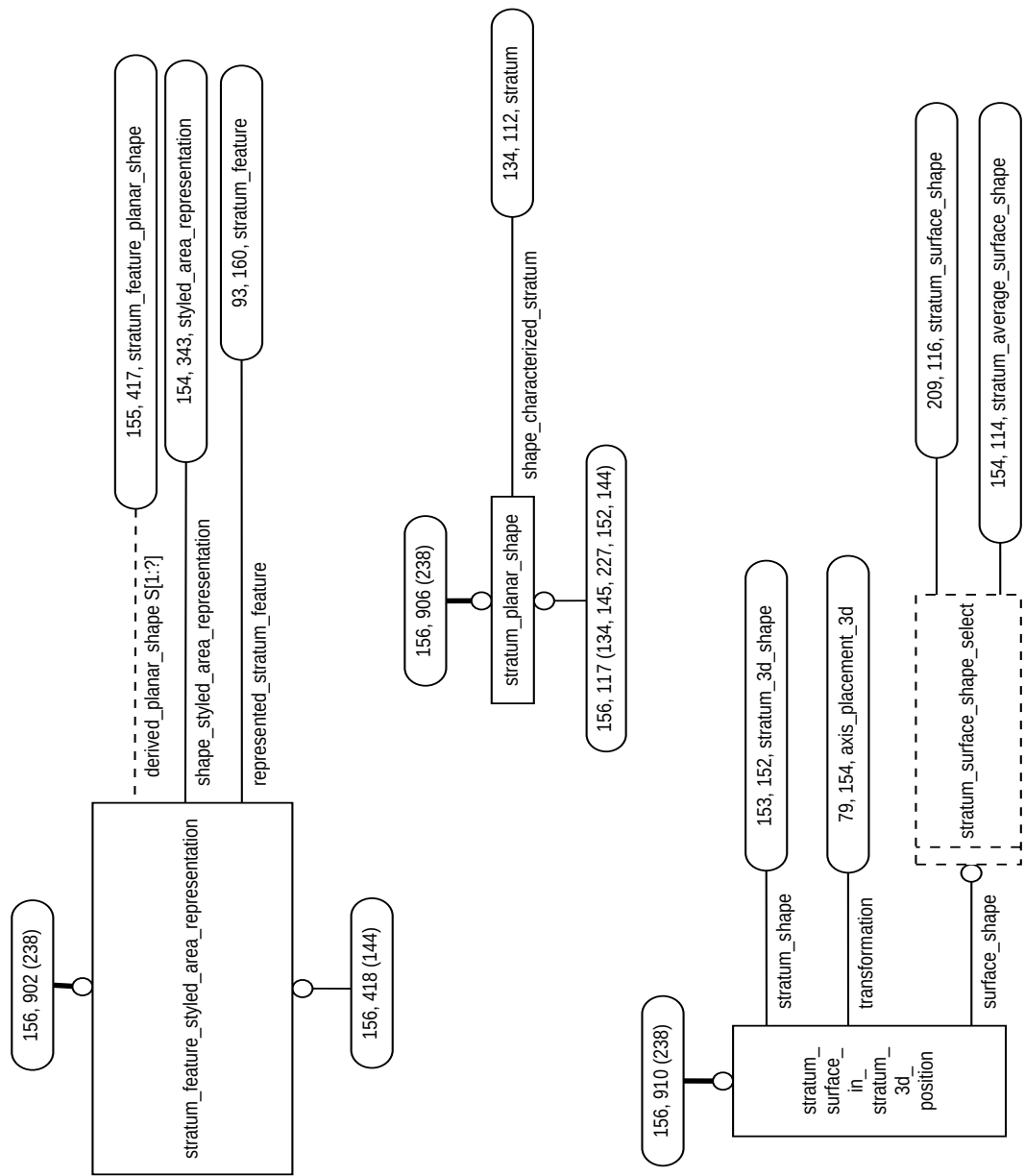


Figure H.156 – ARM EXPRESS-G Graphic 156 of 243

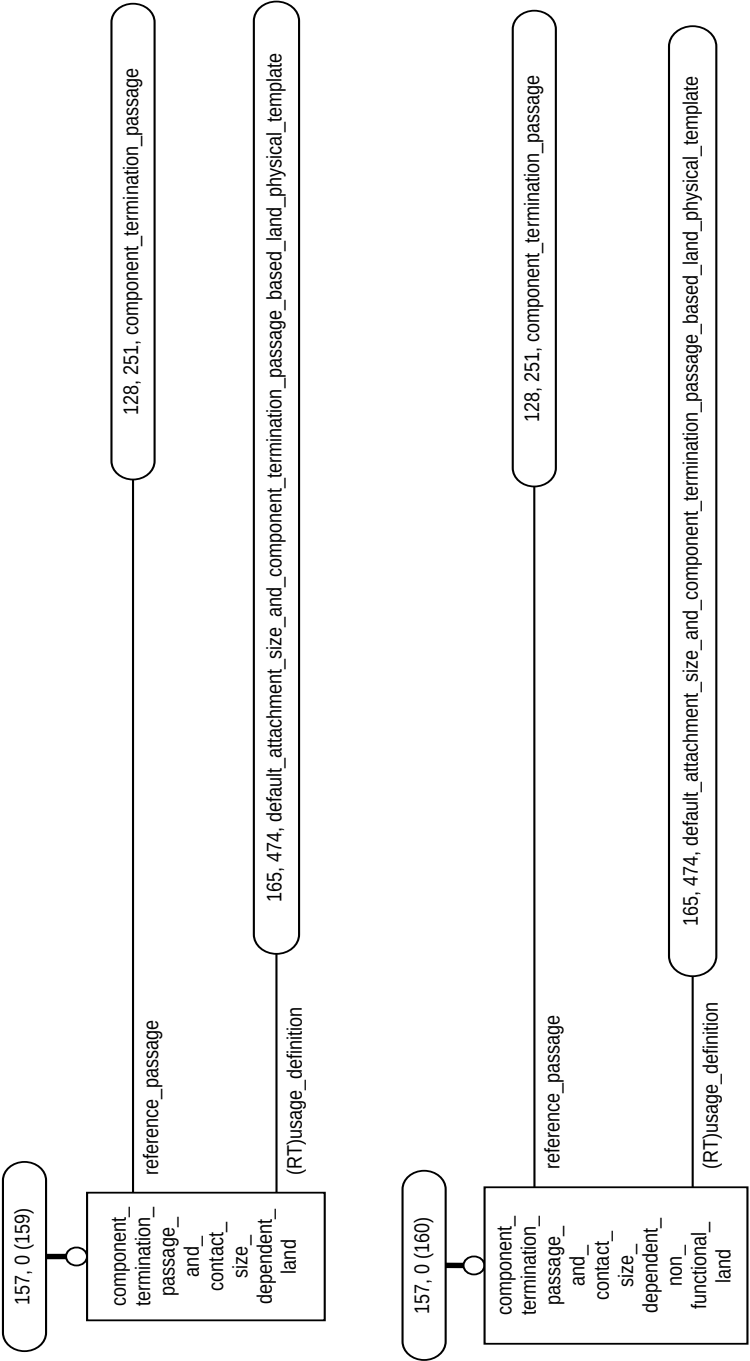


Figure H.157 – ARM EXPRESS-G Graphic 157 of 243

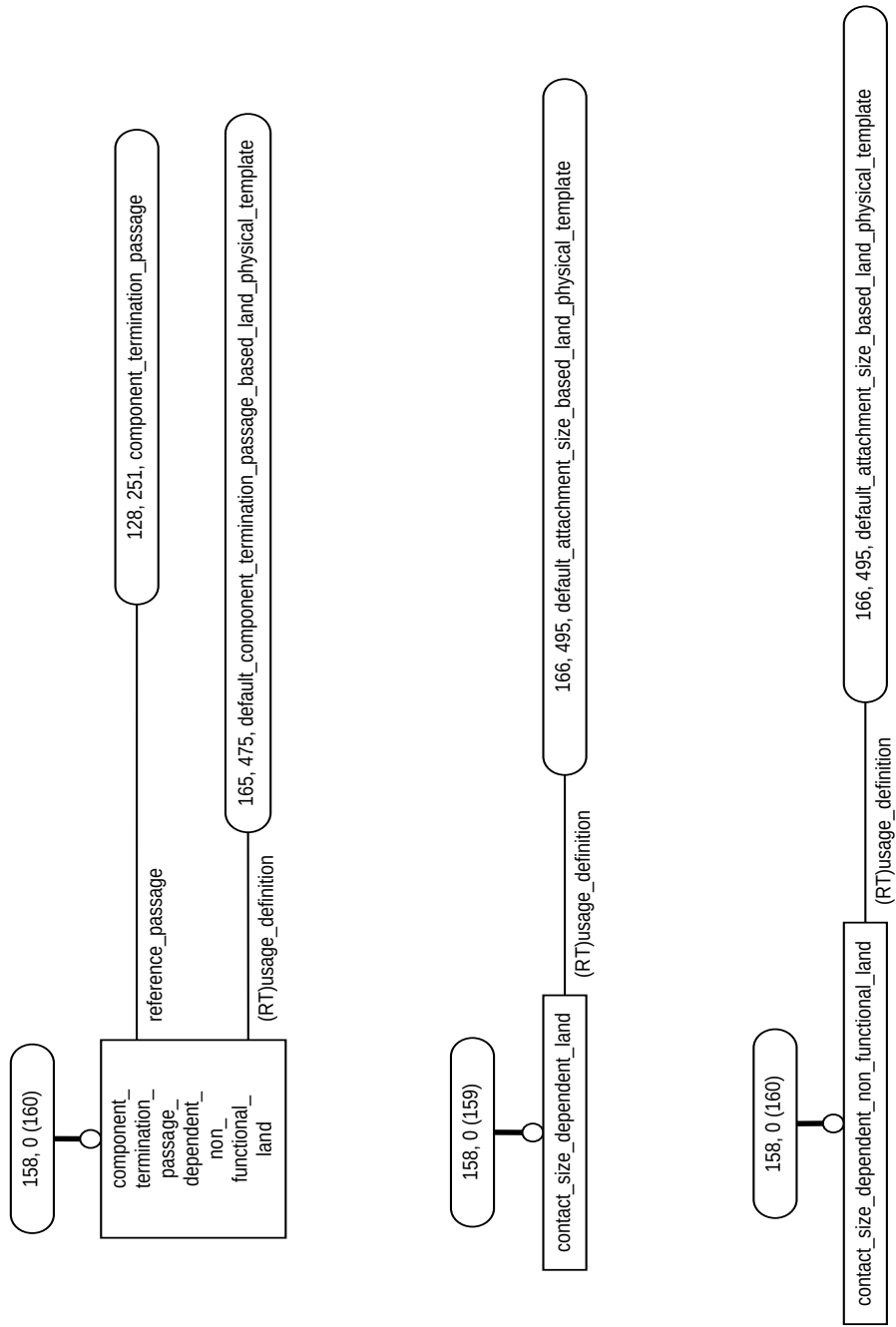


Figure H.158 – ARM EXPRESS-G Graphic 158 of 243

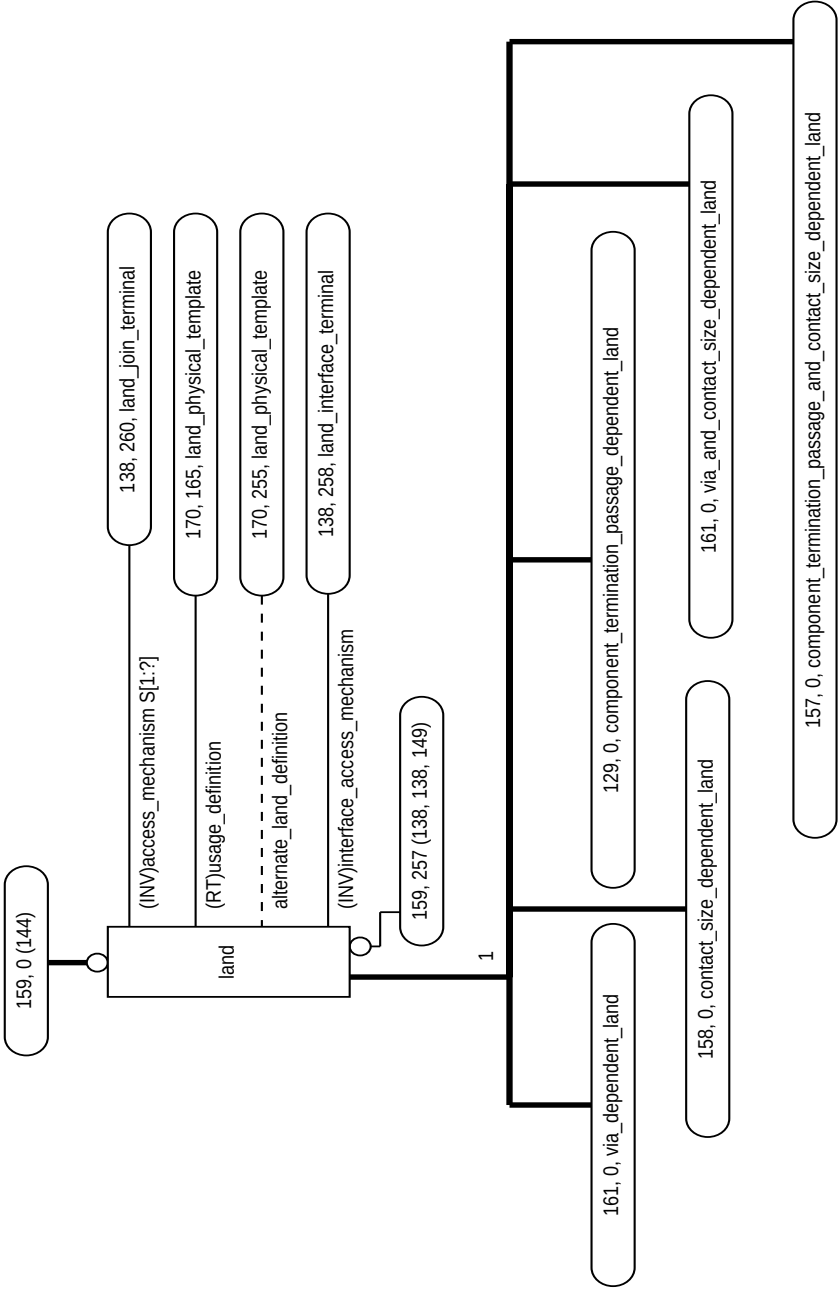


Figure H.159 – ARM EXPRESS-G Graphic 159 of 243

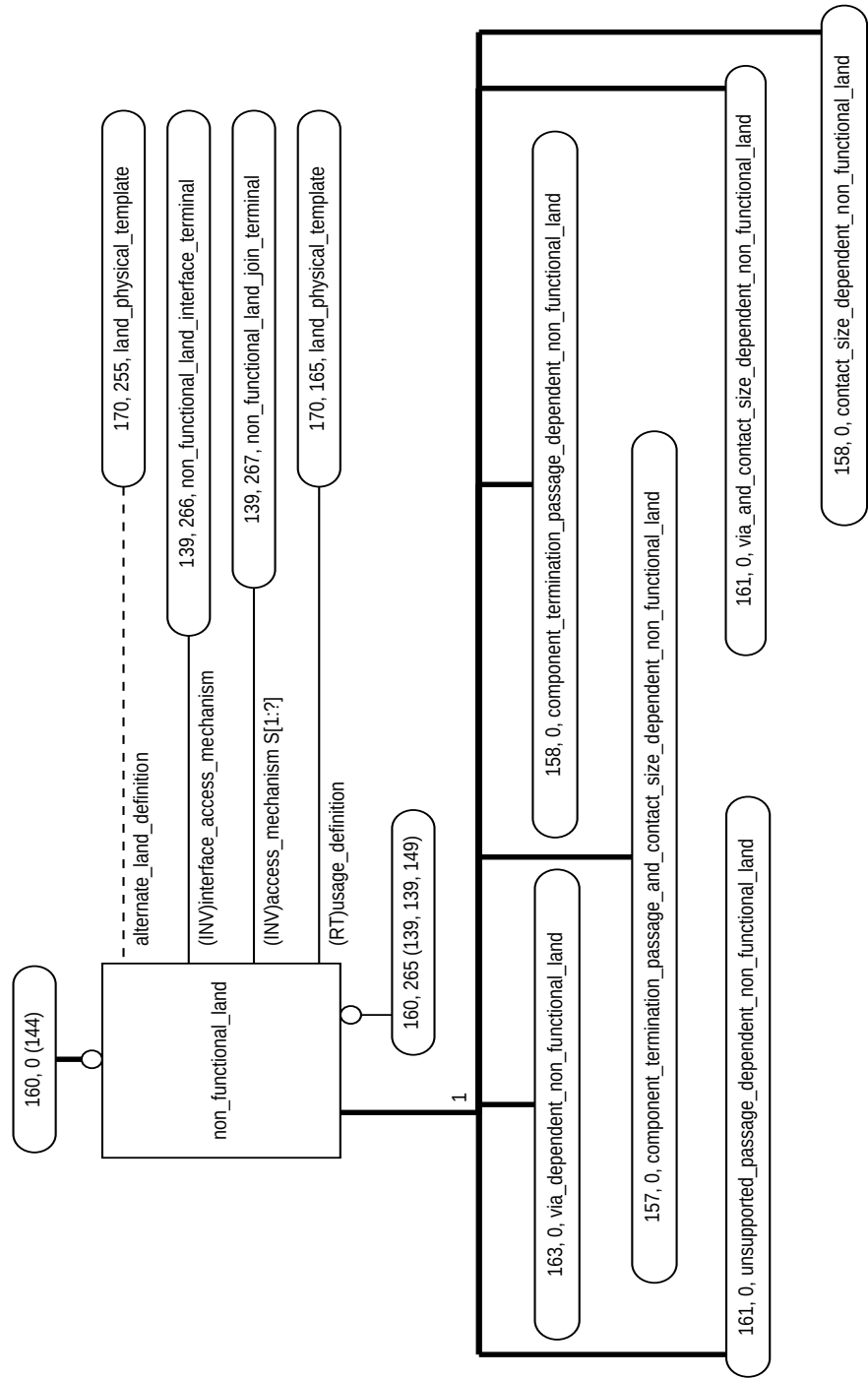


Figure H.160 – ARM EXPRESS-G Graphic 160 of 243

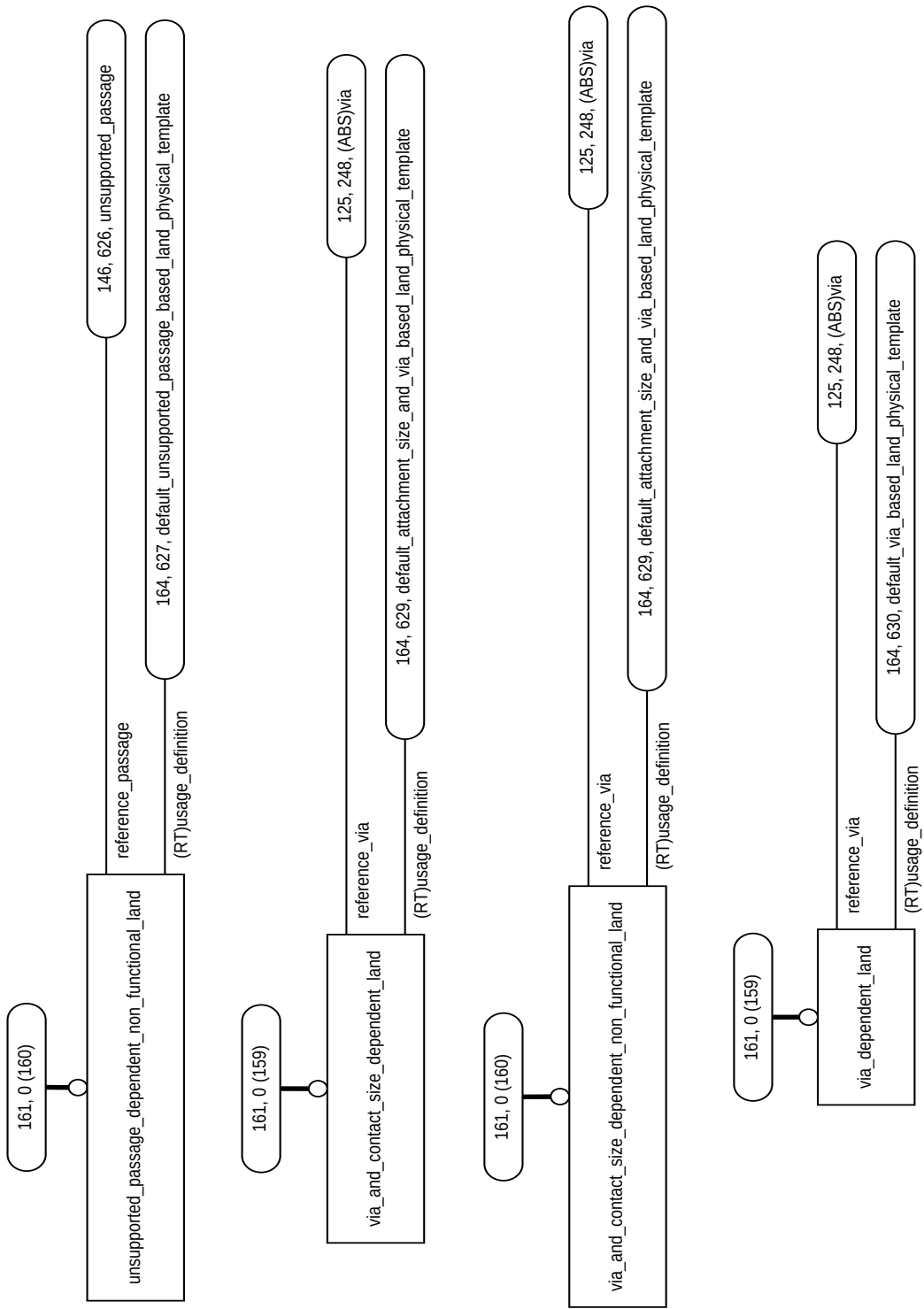


Figure H.161 – ARM EXPRESS-G Graphic 161 of 243

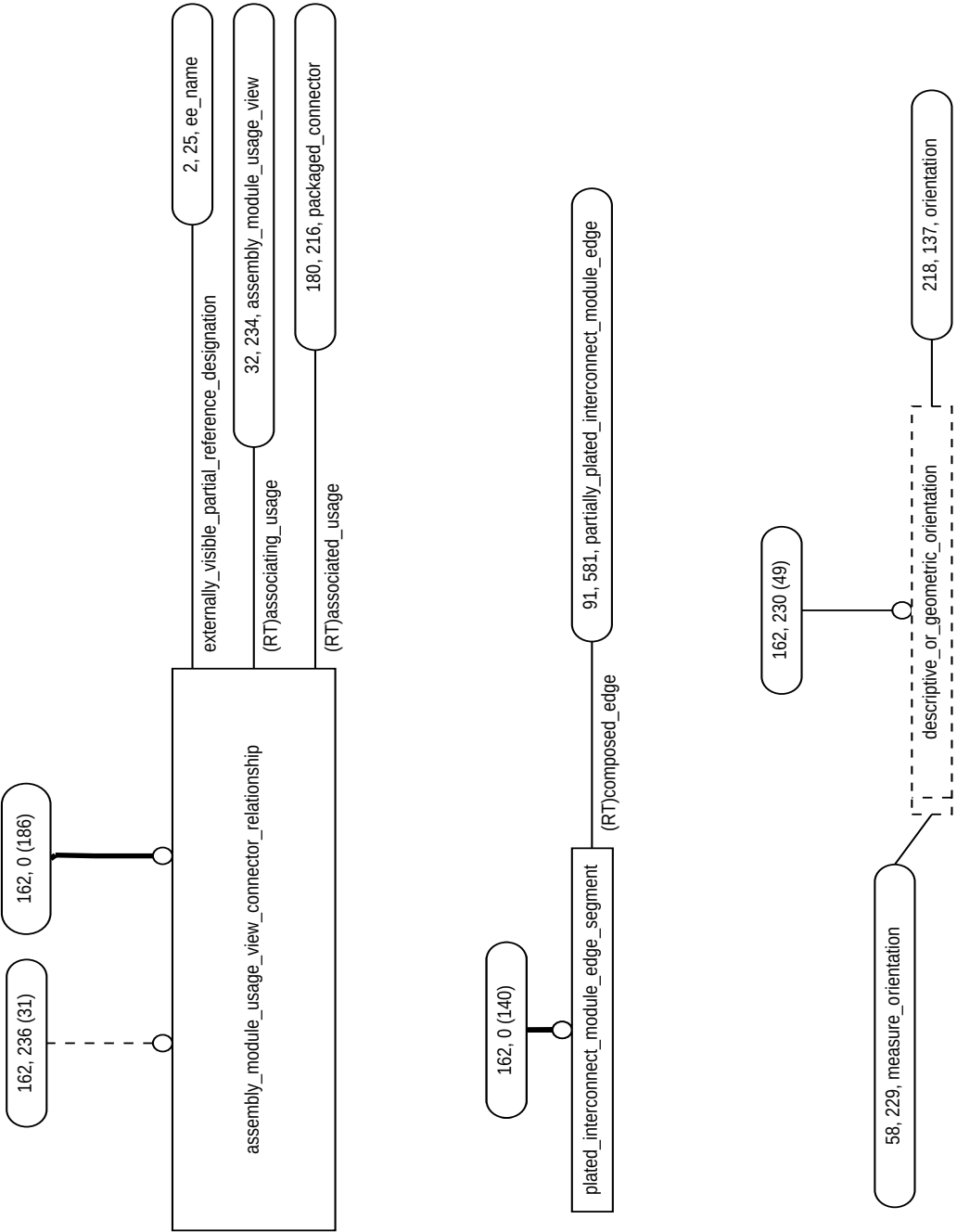


Figure H.162 – ARM EXPRESS-G Graphic 162 of 243

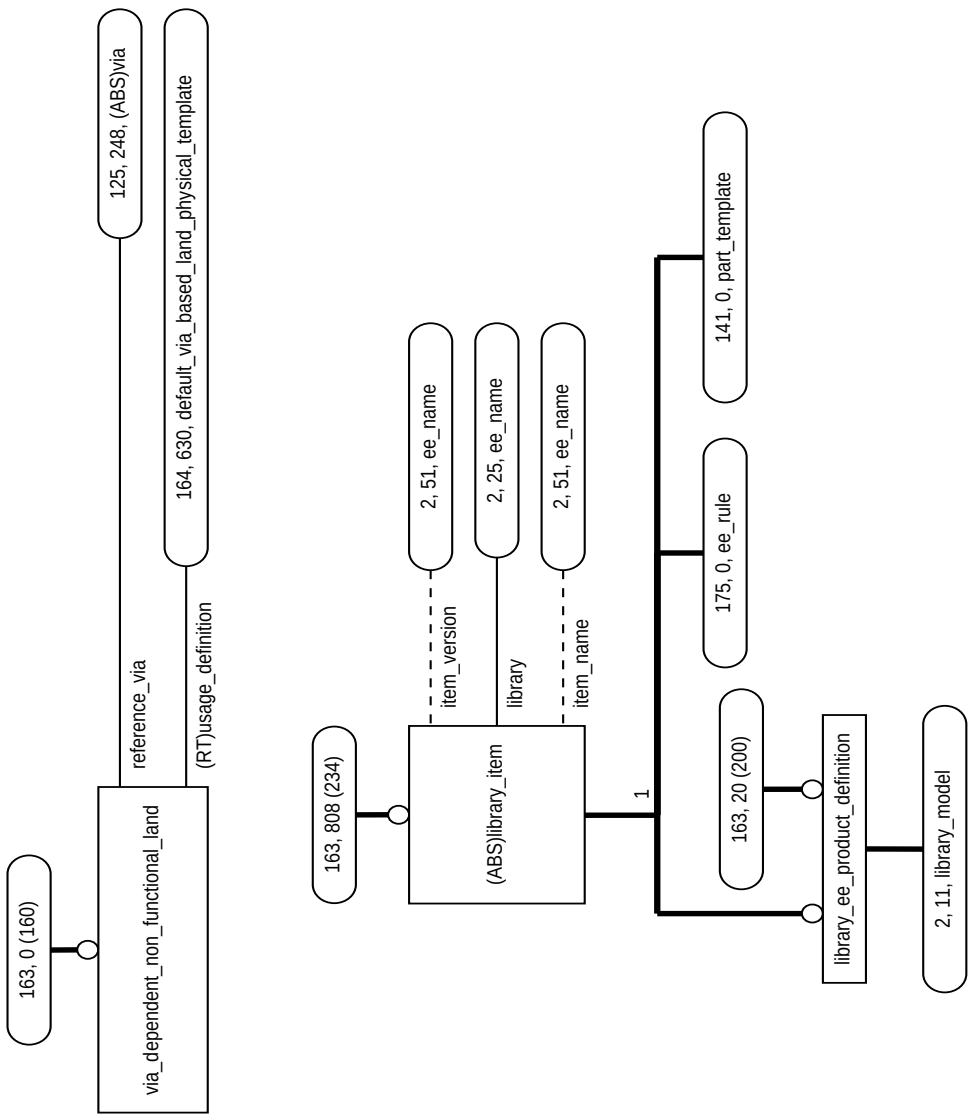


Figure H.163 – ARM EXPRESS-G Graphic 163 of 243

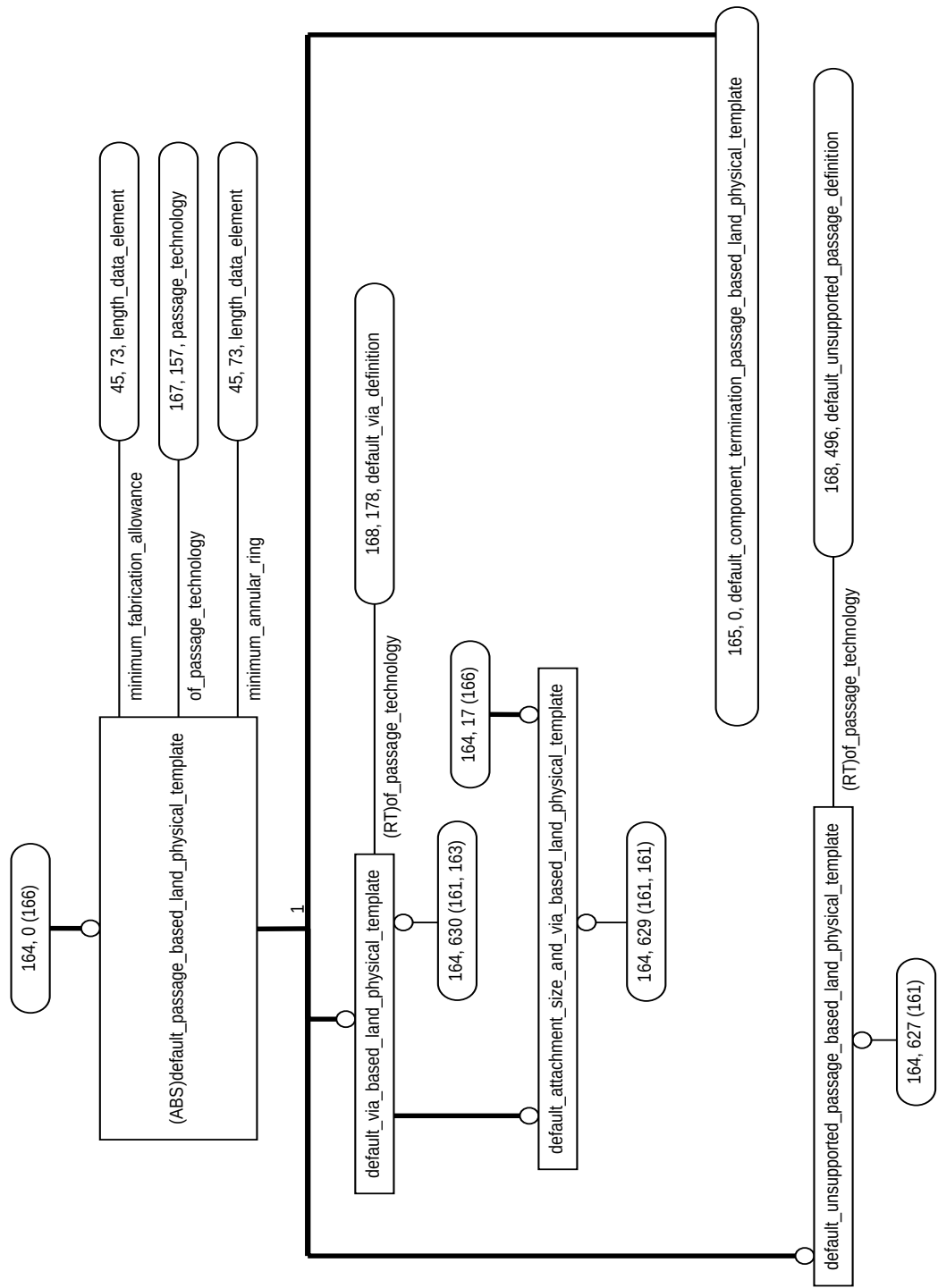


Figure H.164 – ARM EXPRESS-G Graphic 164 of 243

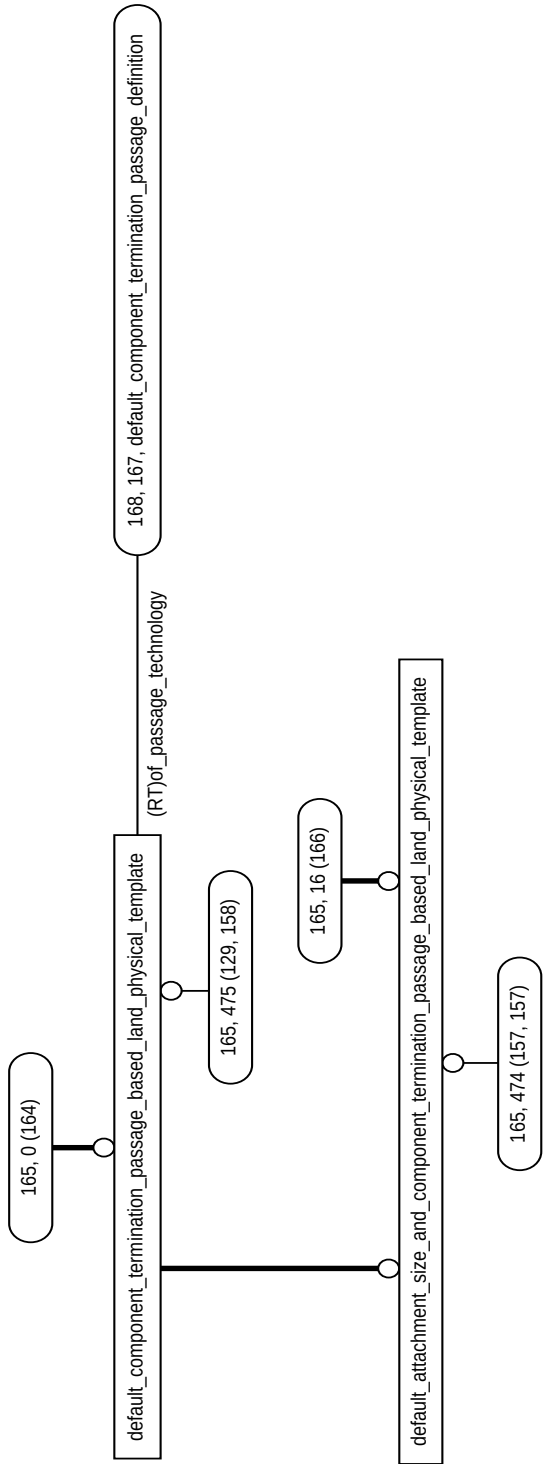


Figure H.165 – ARM EXPRESS-G Graphic 165 of 243

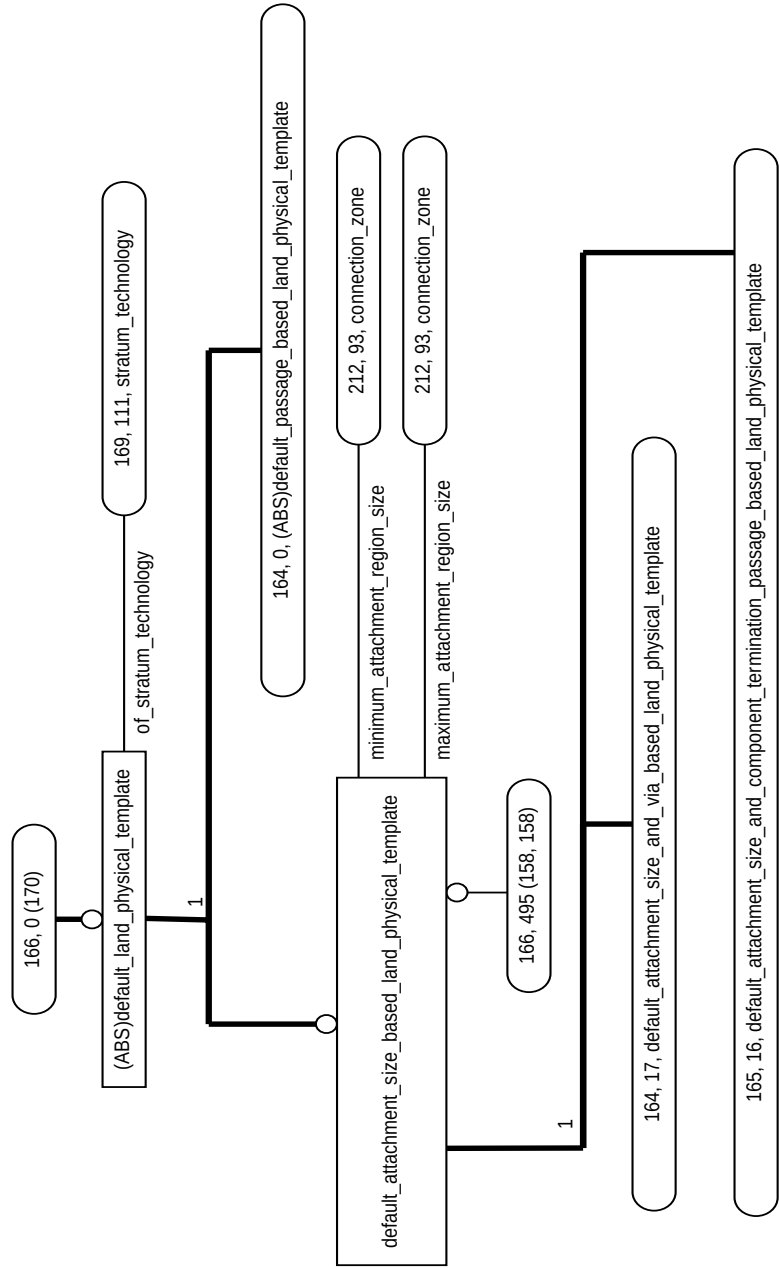


Figure H.166 – ARM EXPRESS-G Graphic 166 of 243

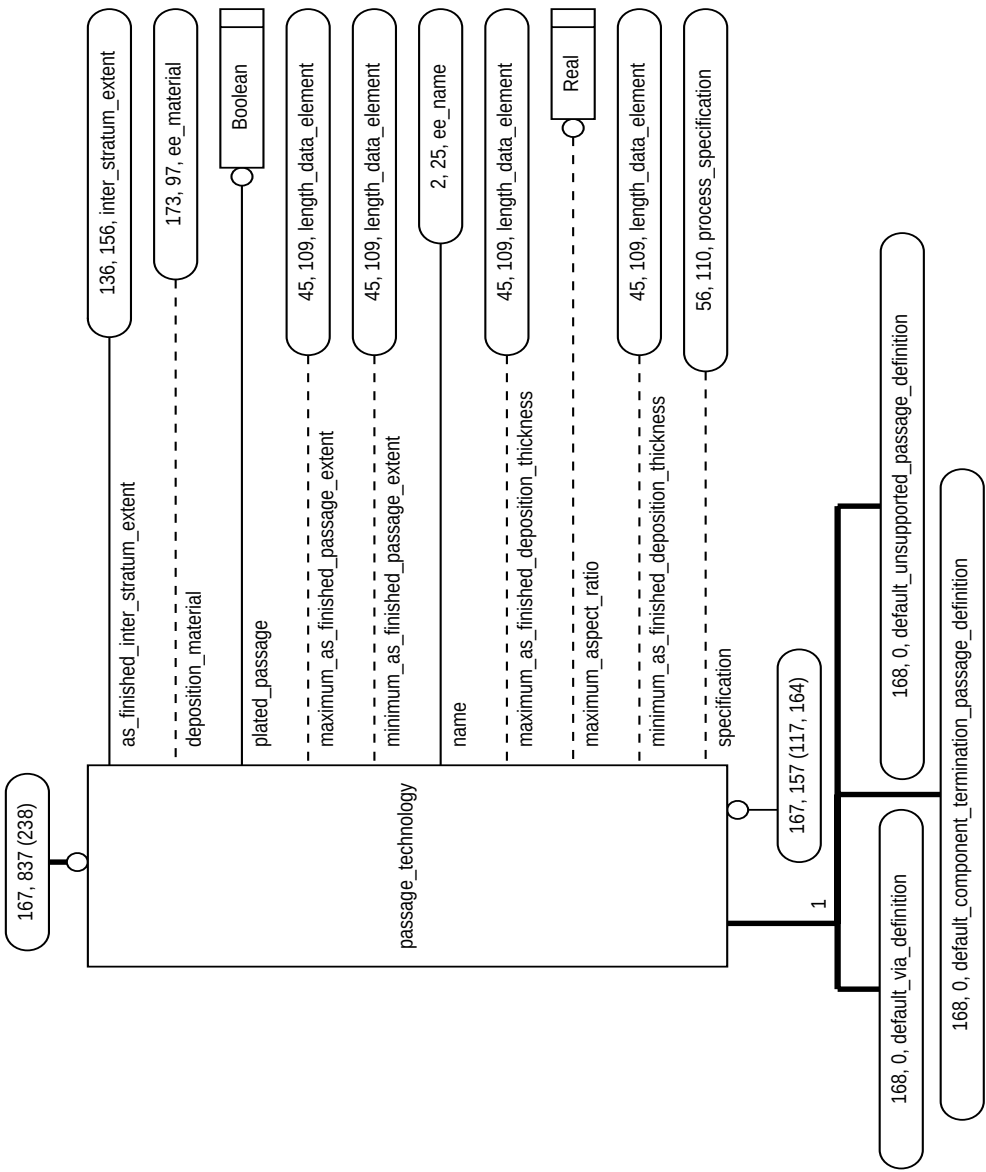


Figure H.167 – ARM EXPRESS-G Graphic 167 of 243

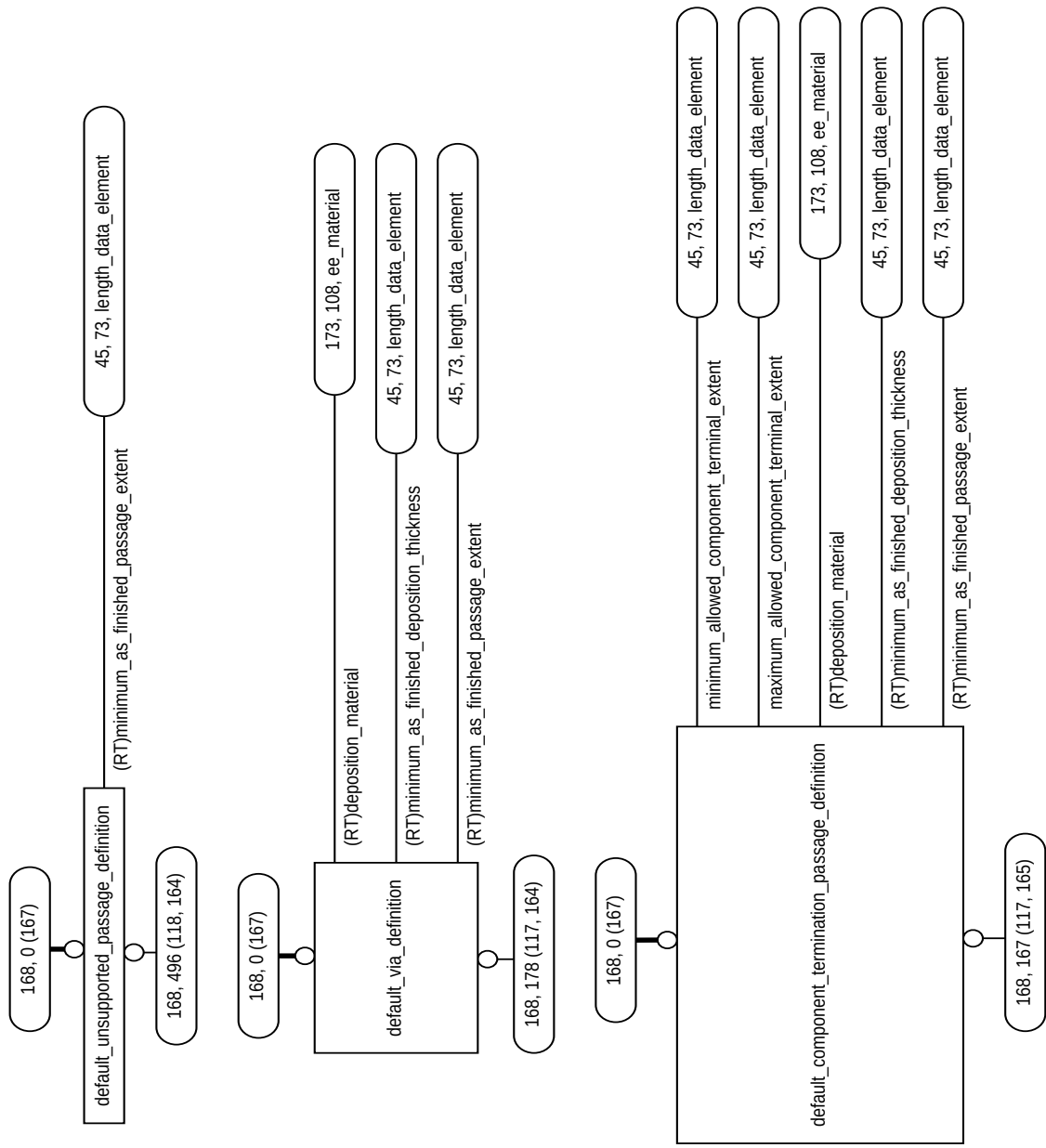


Figure H.168 – ARM EXPRESS-G Graphic 168 of 243

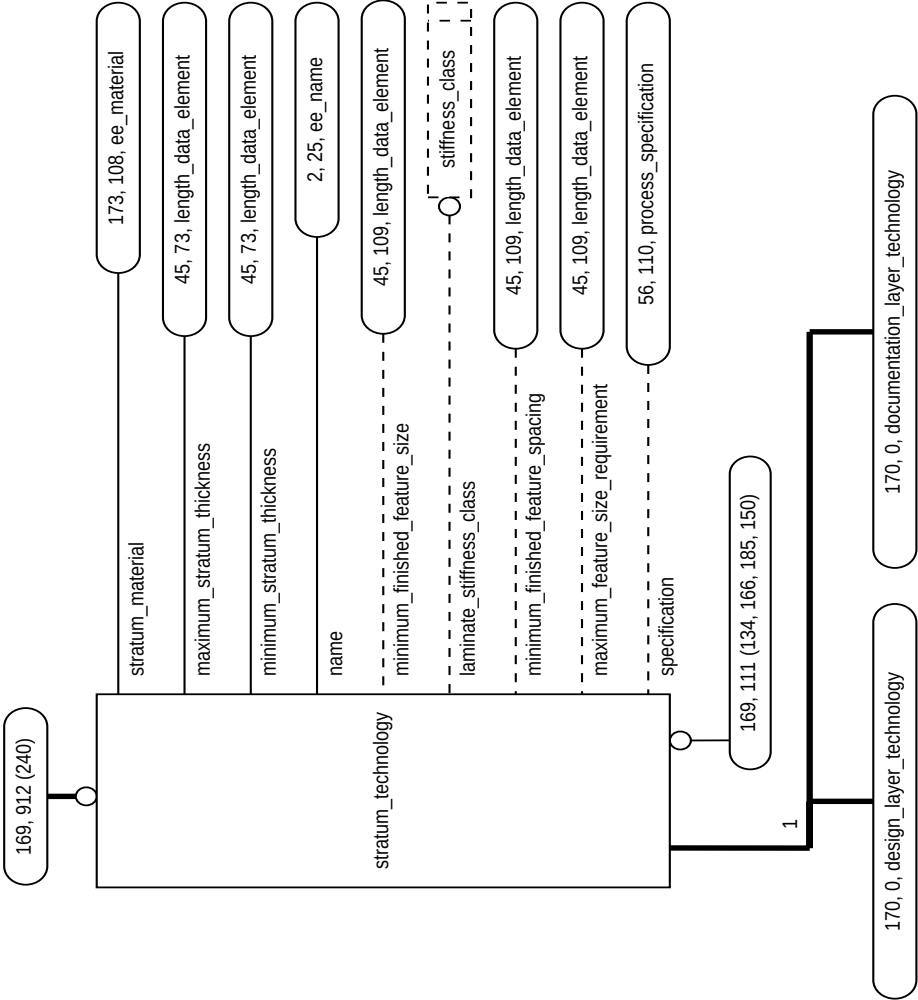


Figure H.169 – ARM EXPRESS-G Graphic 169 of 243

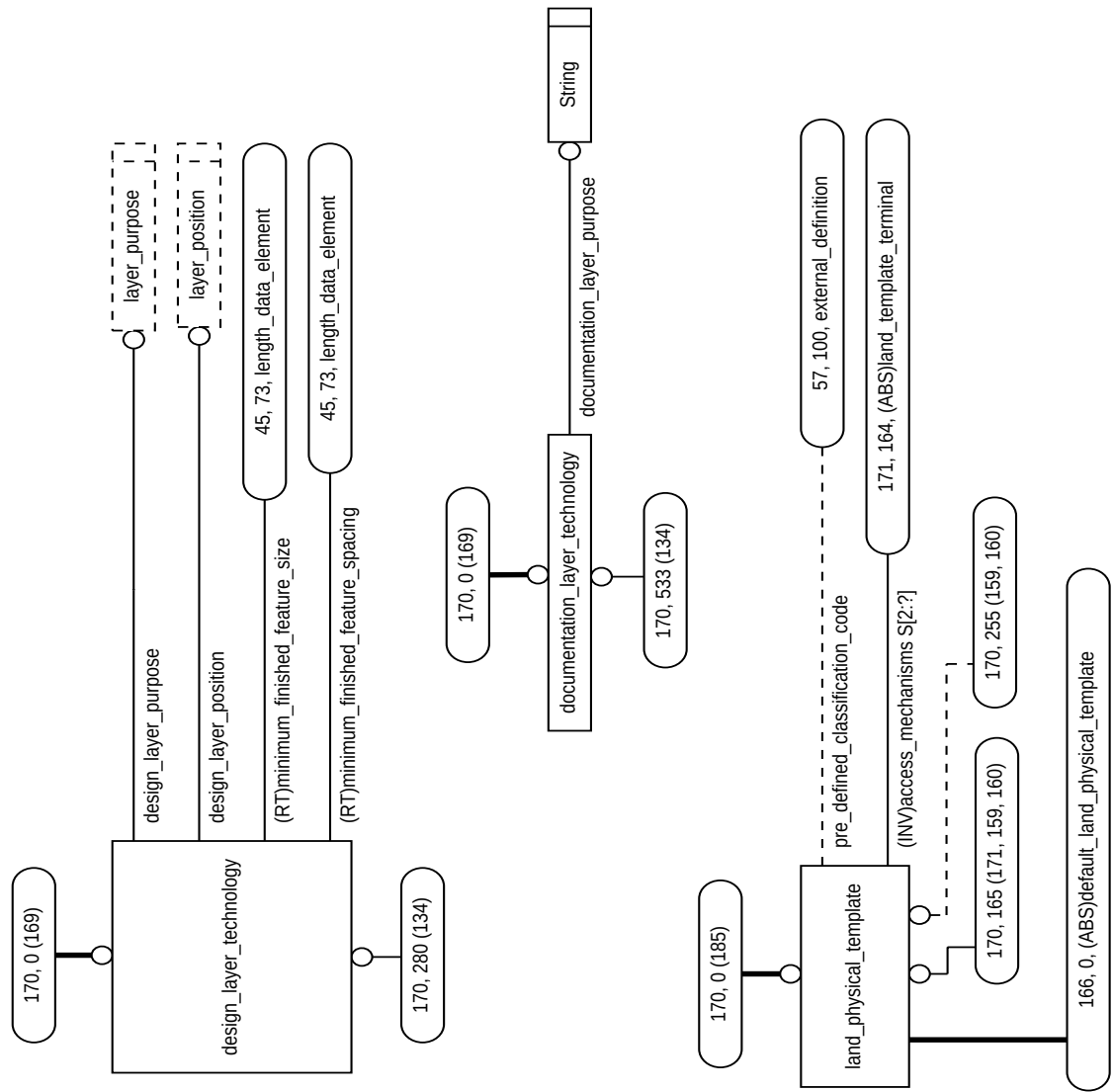


Figure H.170 – ARM EXPRESS-G Graphic 170 of 243

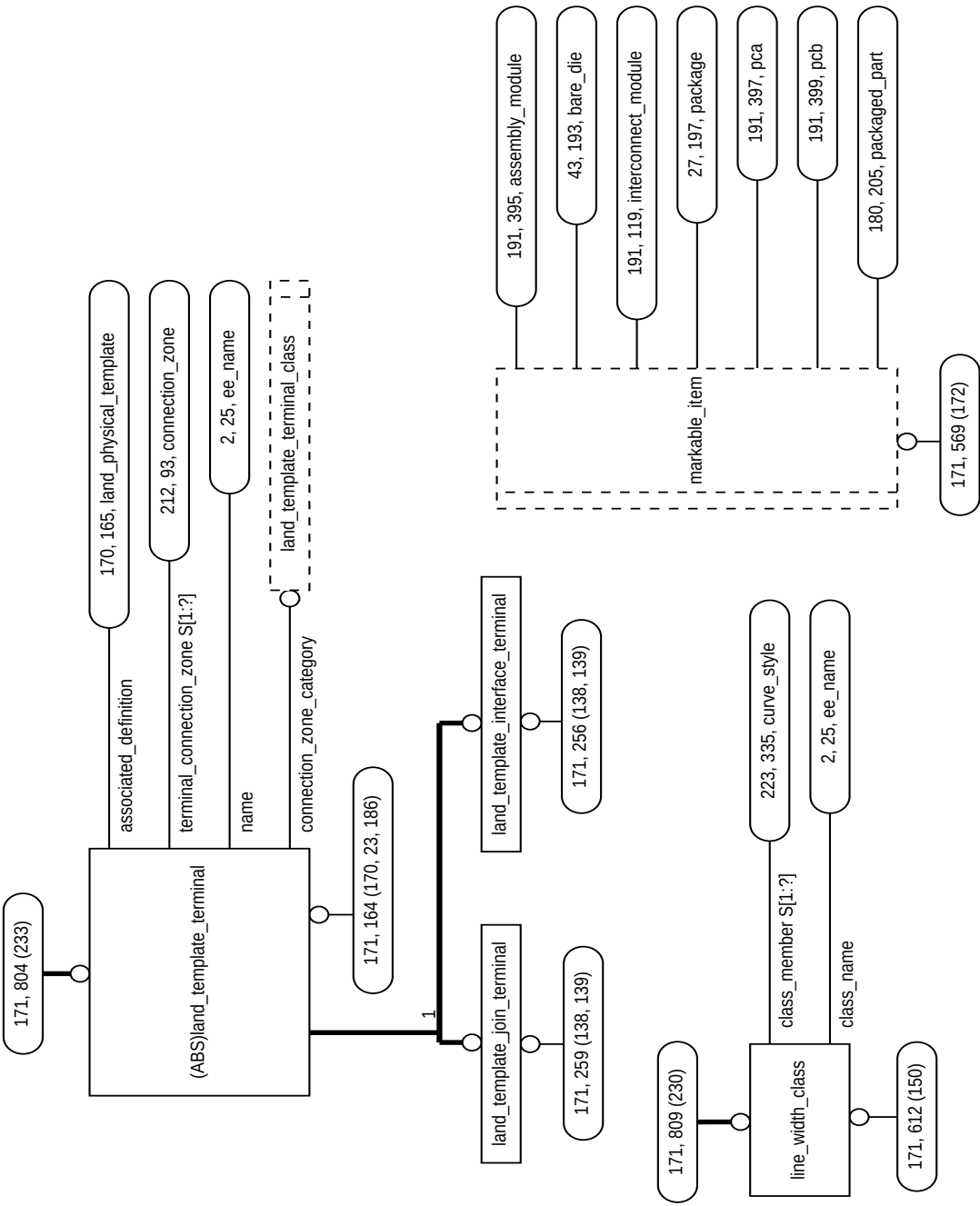


Figure H.171 – ARM EXPRESS-G Graphic 171 of 243

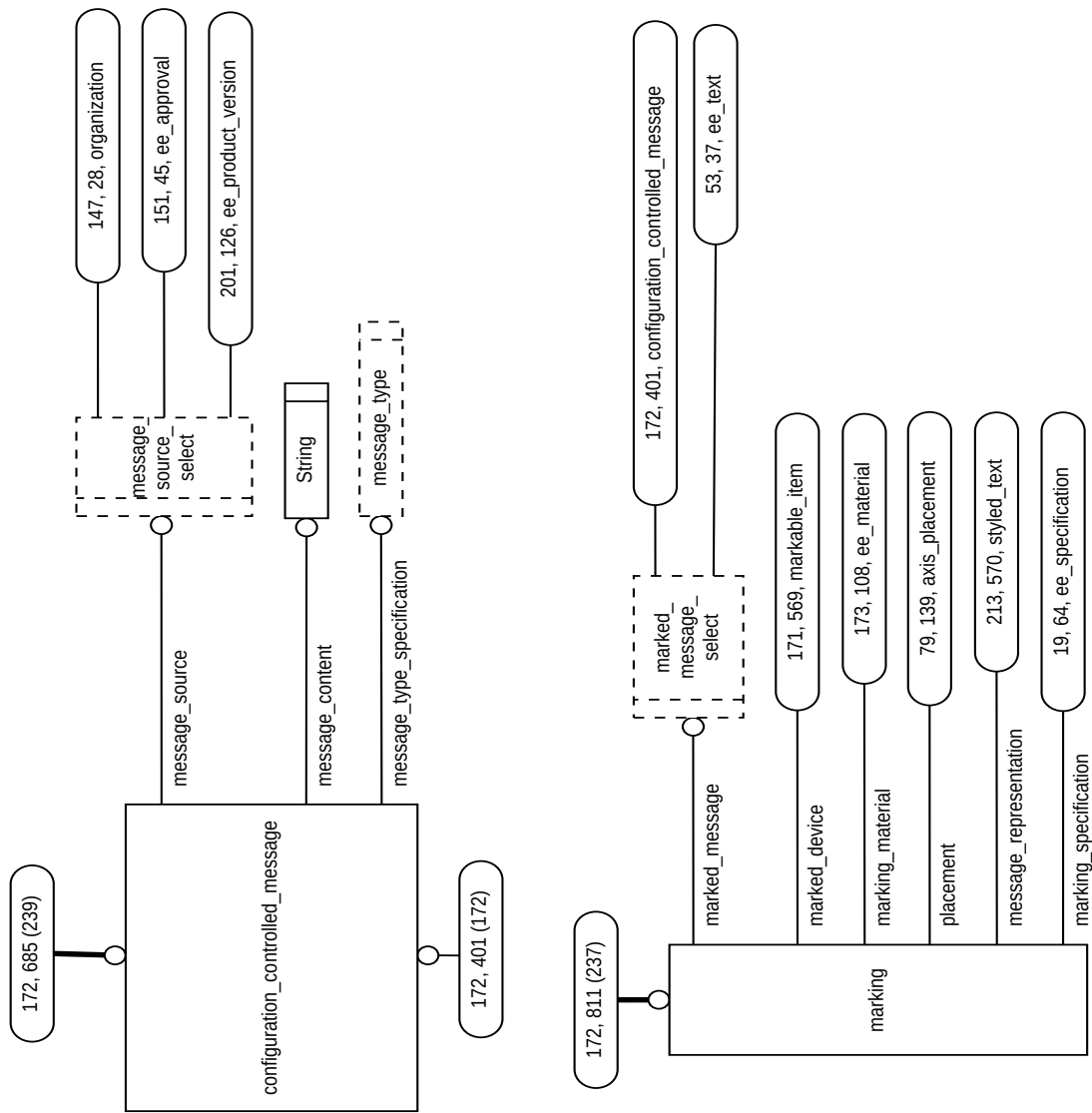


Figure H.172 – ARM EXPRESS-G Graphic 172 of 243

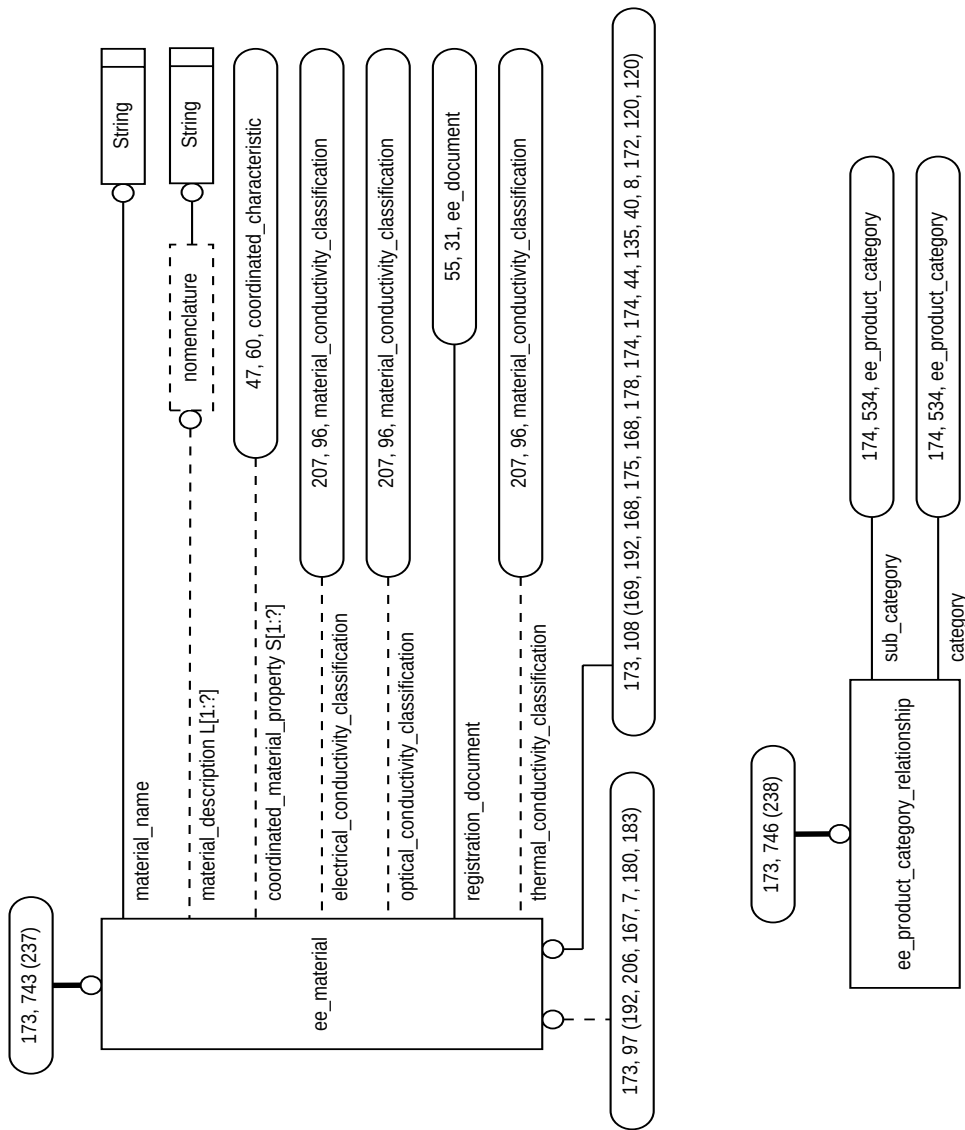


Figure H.173 – ARM EXPRESS-G Graphic 173 of 243

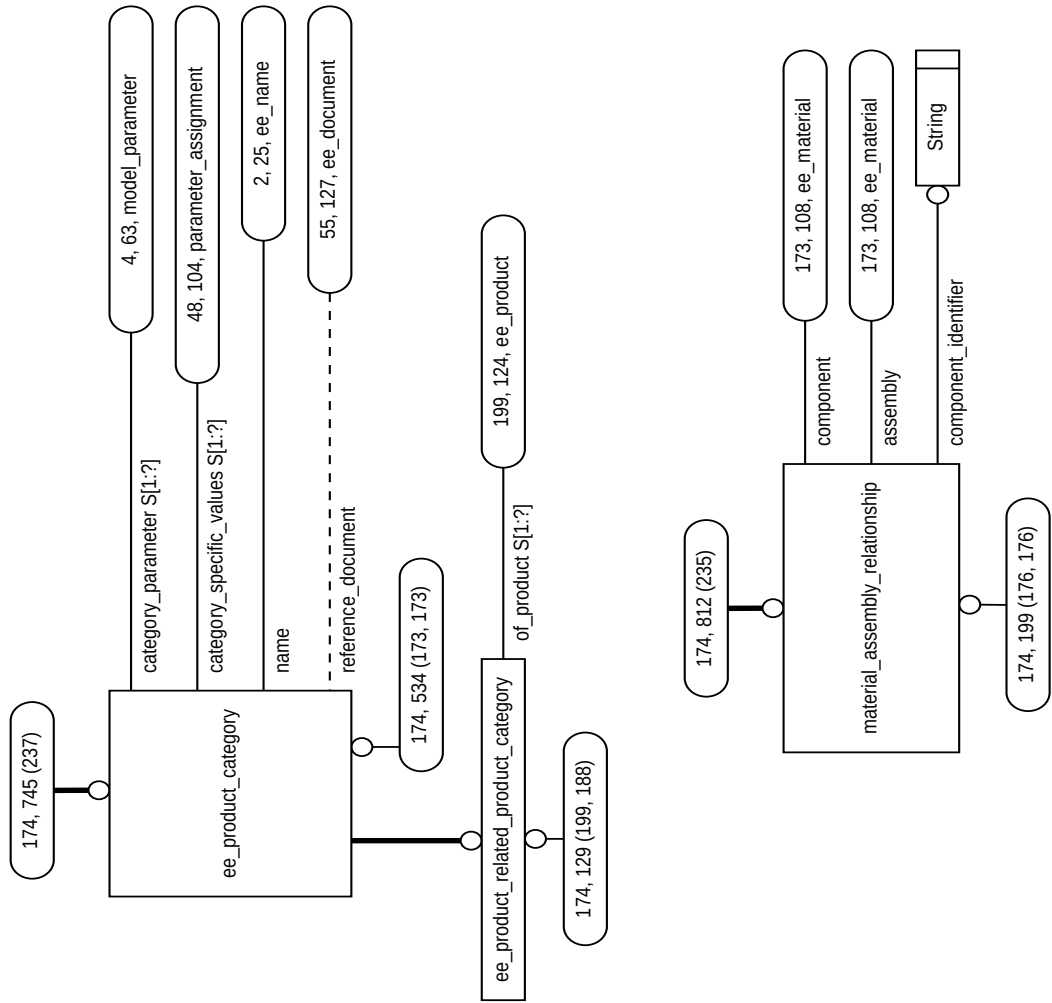


Figure H.174 – ARM EXPRESS-G Graphic 174 of 243

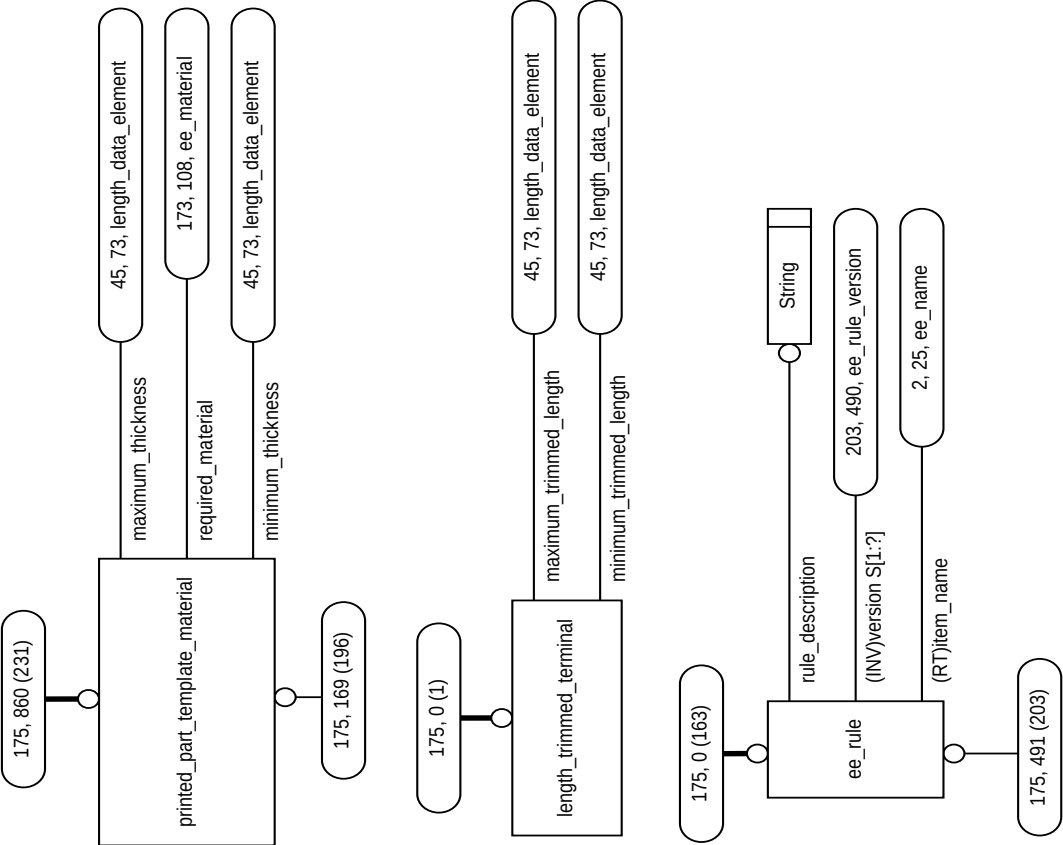


Figure H.175 – ARM EXPRESS-G Graphic 175 of 243

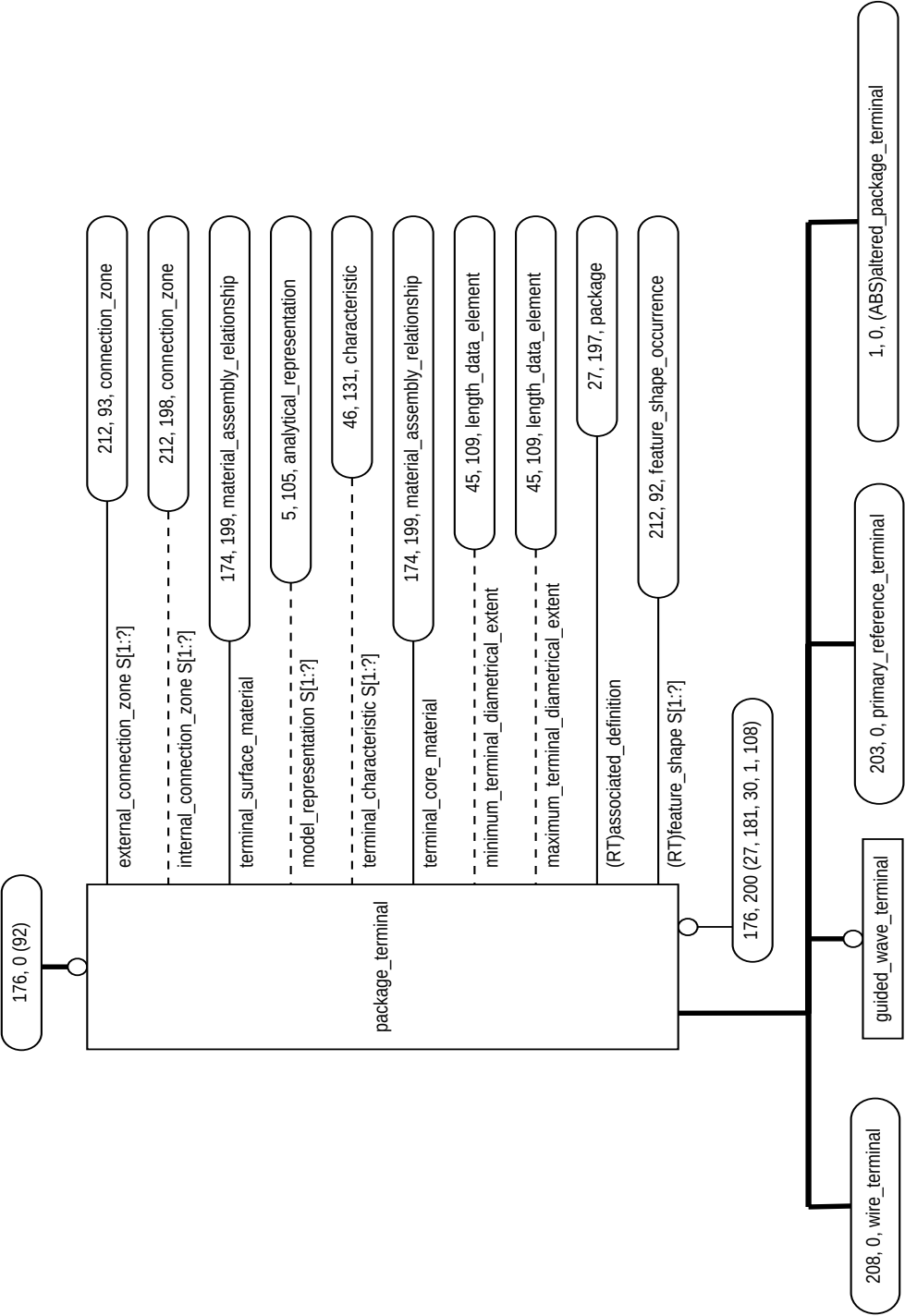


Figure H.176 – ARM EXPRESS-G Graphic 176 of 243

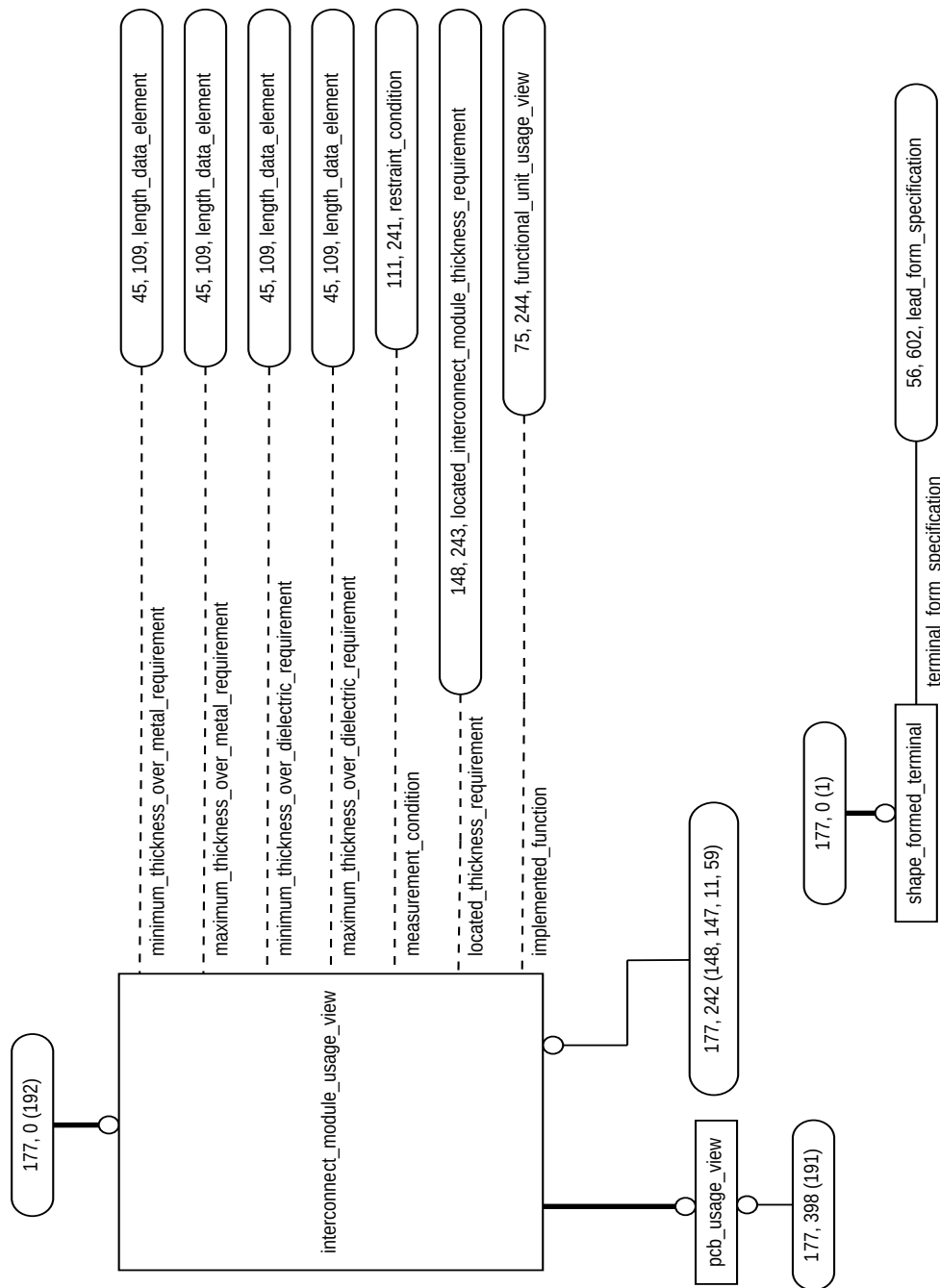


Figure H.177 – ARM EXPRESS-G Graphic 177 of 243

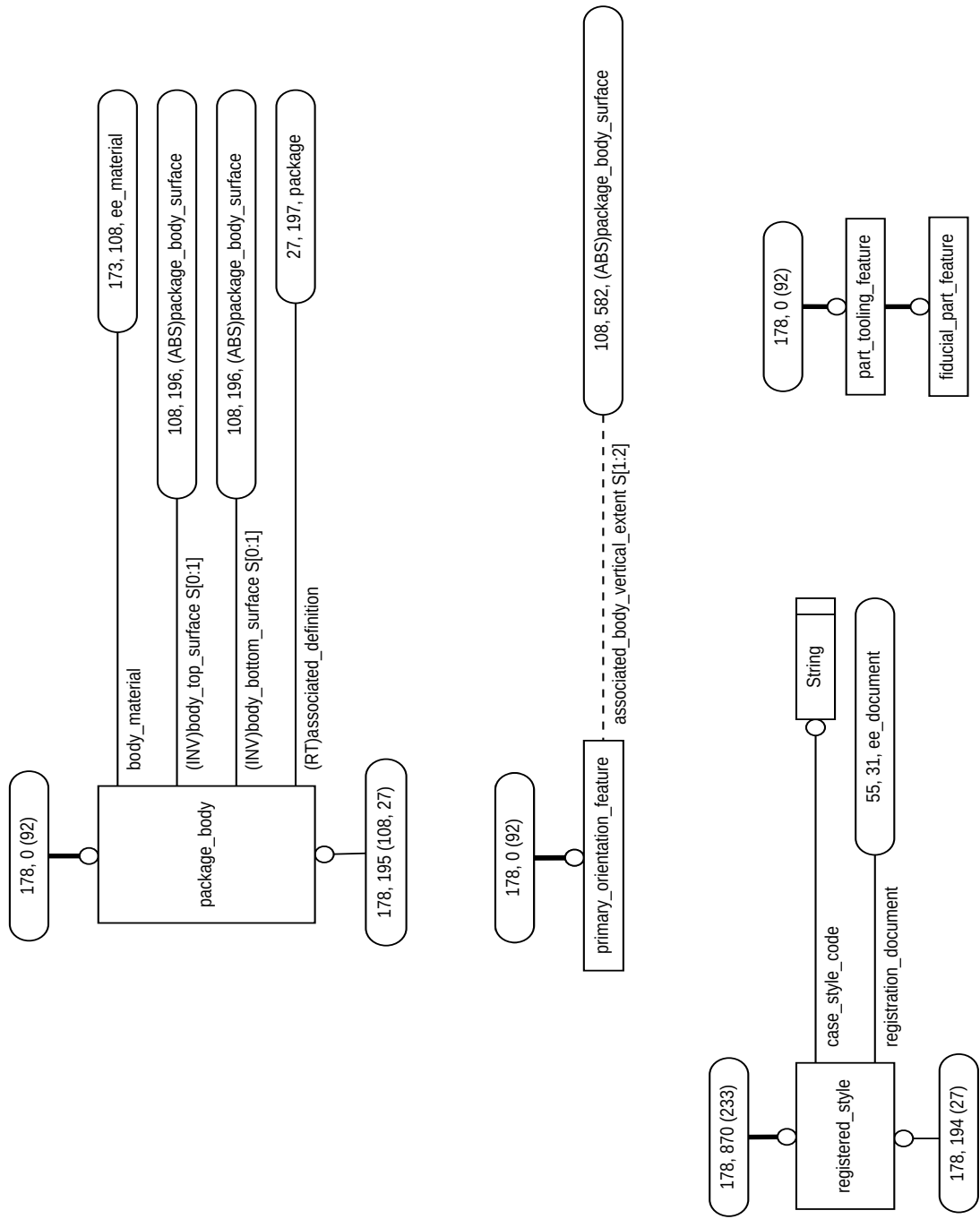


Figure H.178 – ARM EXPRESS-G Graphic 178 of 243

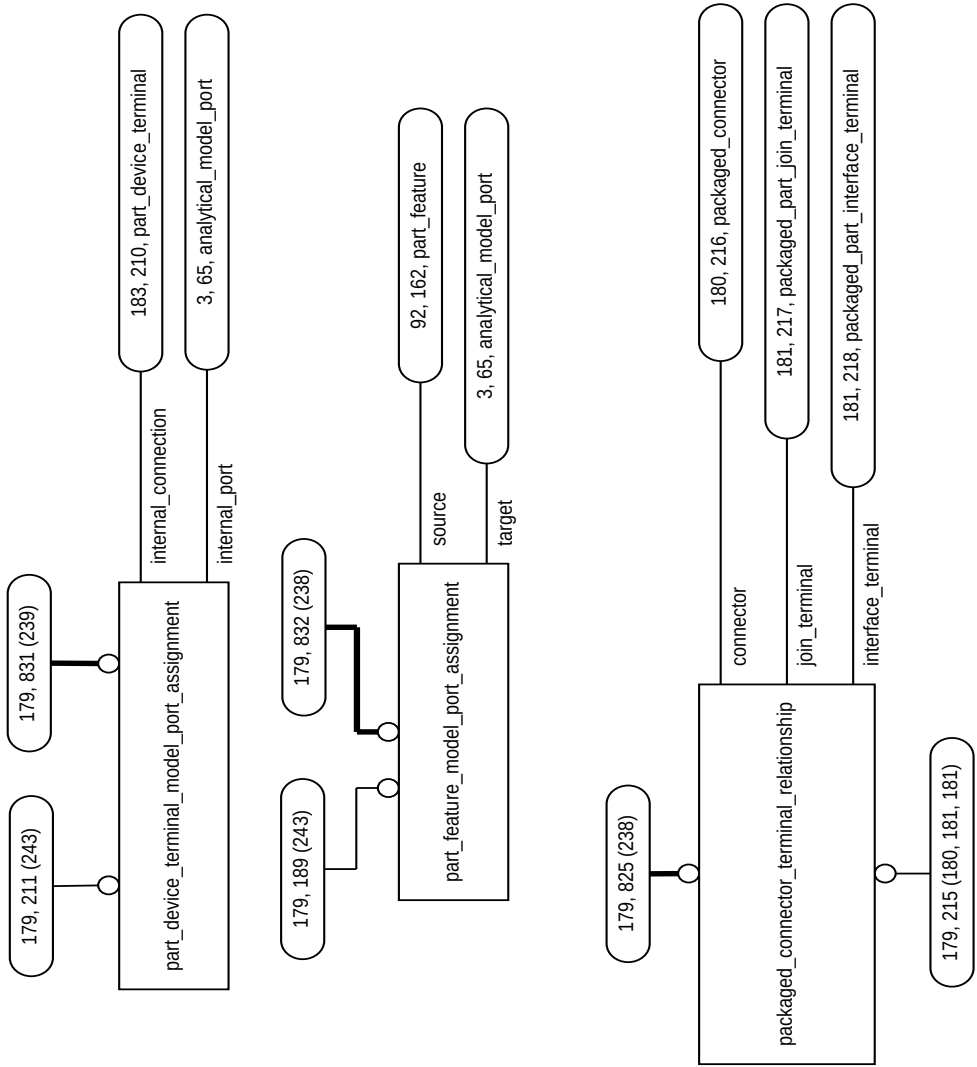


Figure H.179 – ARM EXPRESS-G Graphic 179 of 243

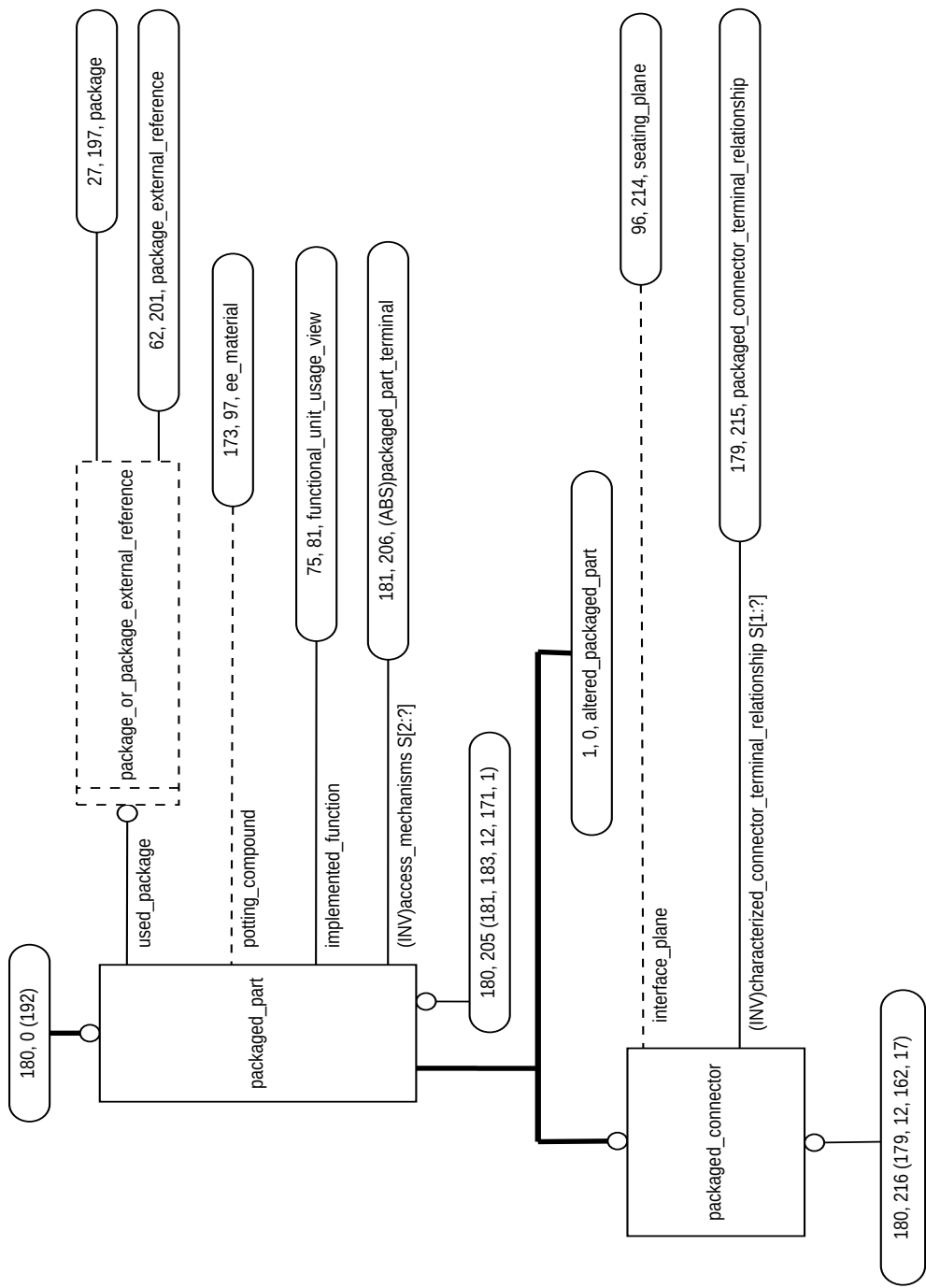


Figure H.180 – ARM EXPRESS-G Graphic 180 of 243

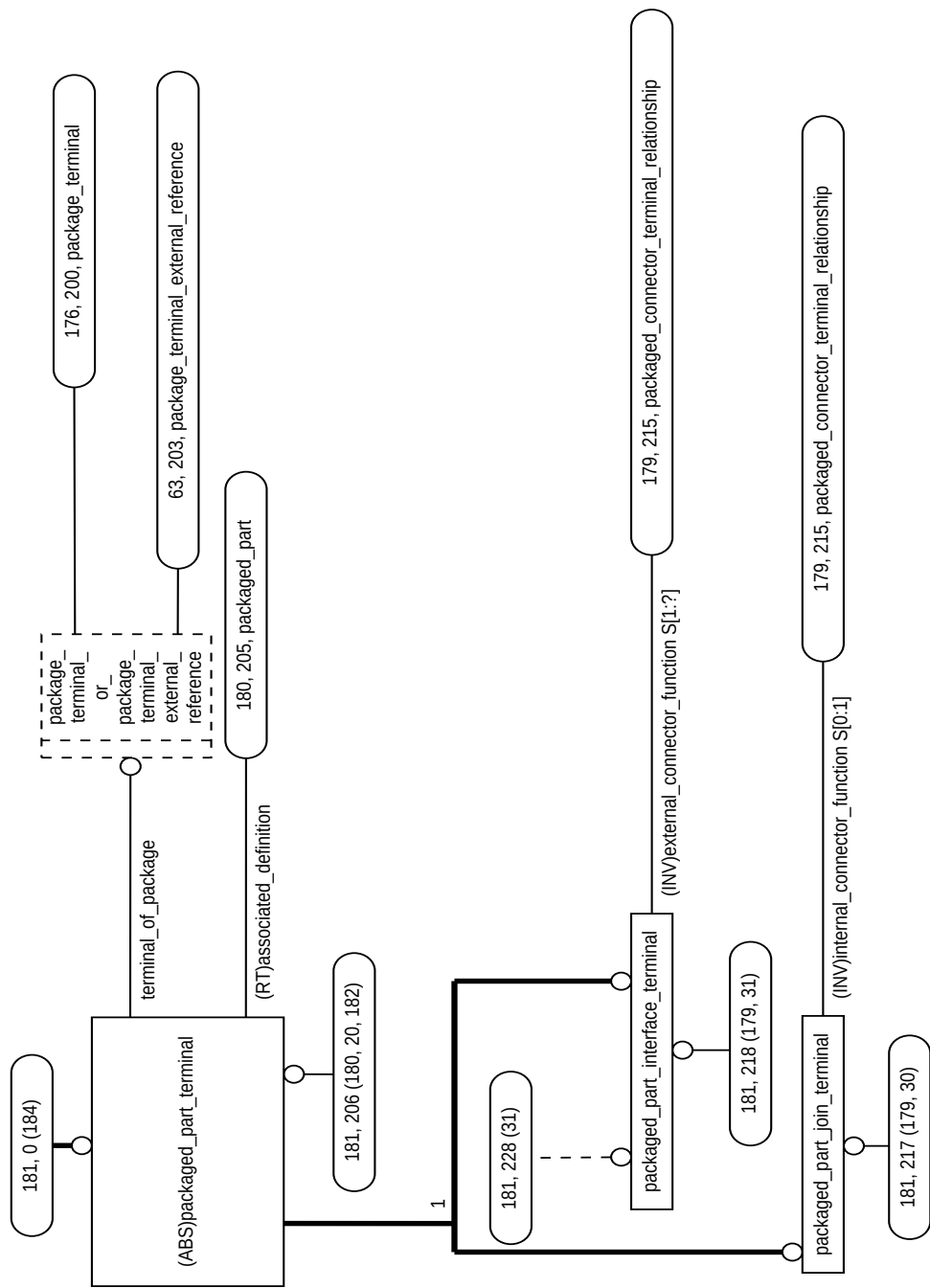


Figure H.181 – ARM EXPRESS-G Graphic 181 of 243

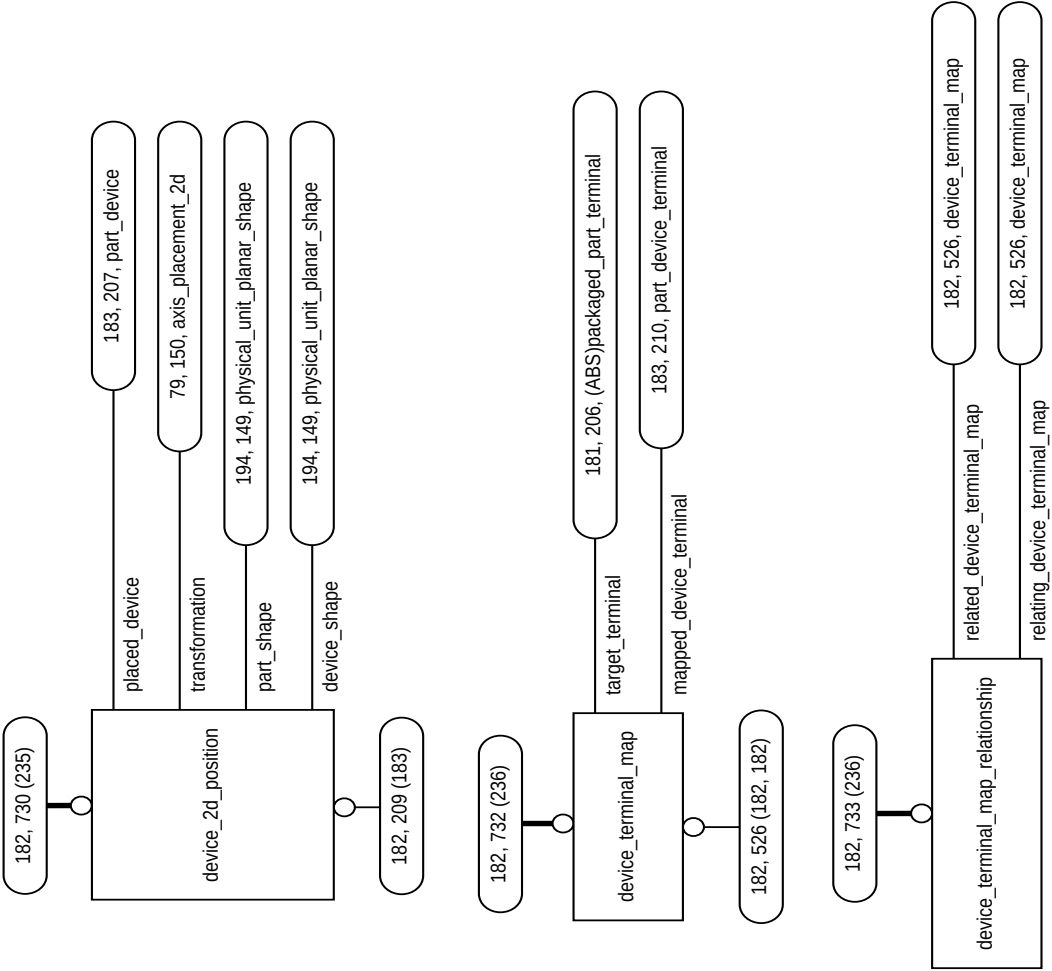


Figure H.182 – ARM EXPRESS-G Graphic 182 of 243

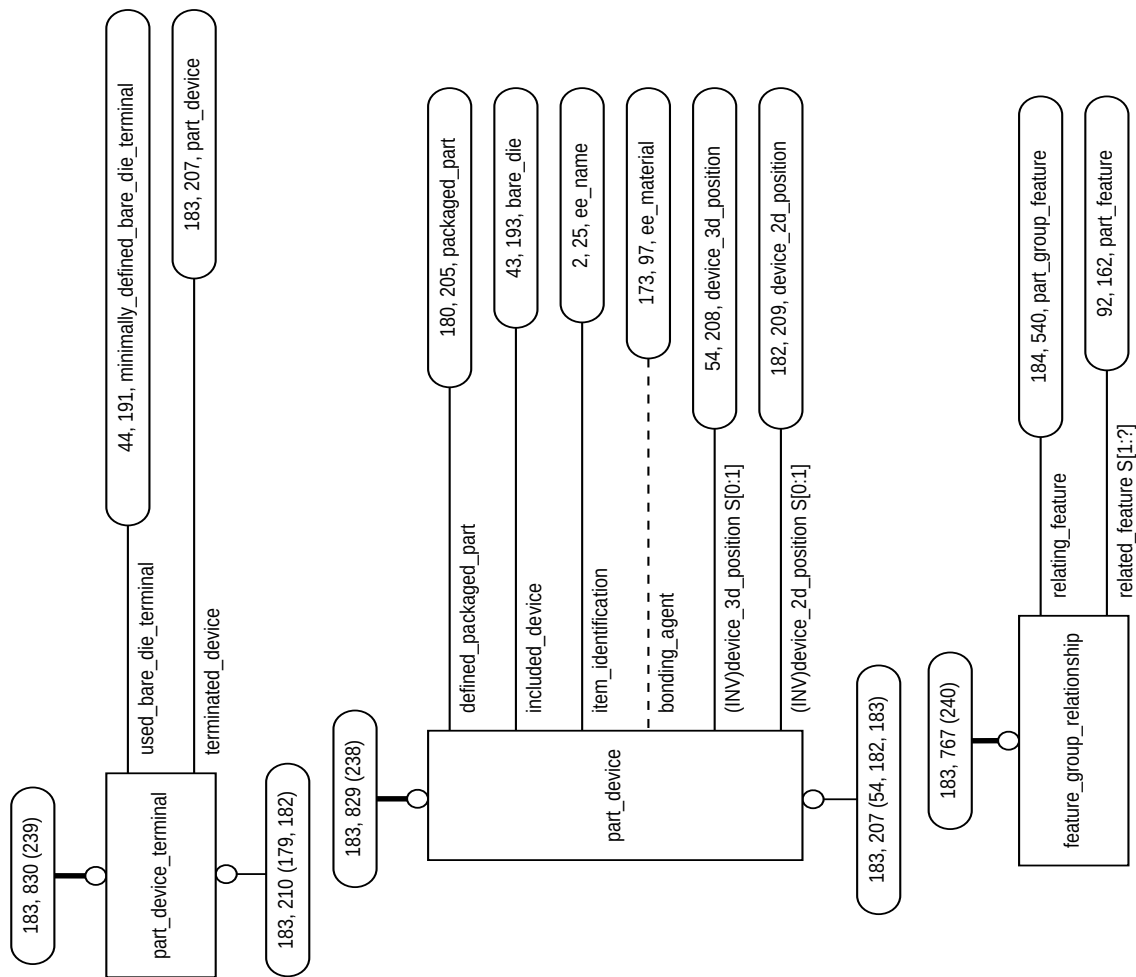


Figure H.183 – ARM EXPRESS-G Graphic 183 of 243

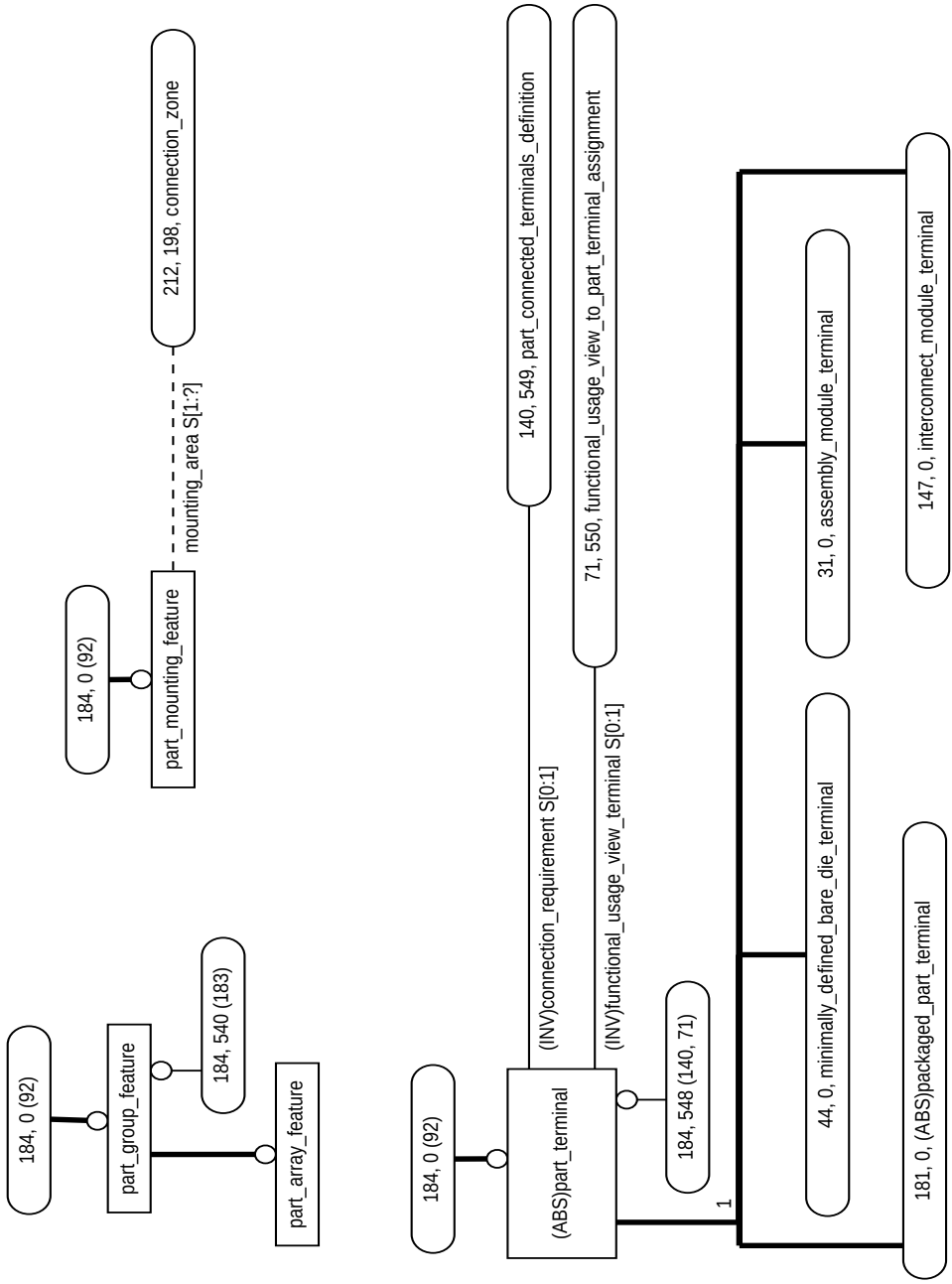


Figure H.184 – ARM EXPRESS-G Graphic 184 of 243

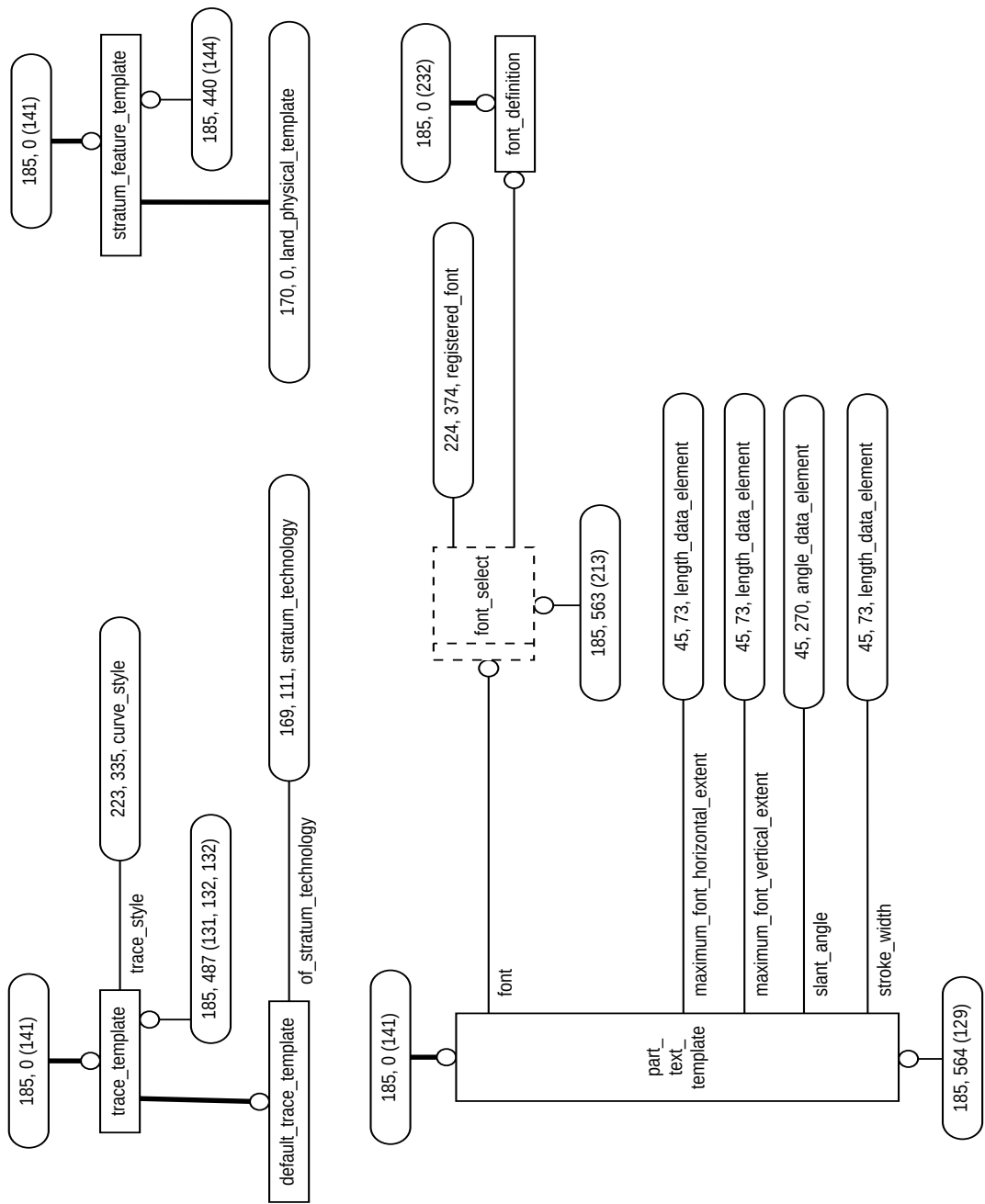


Figure H.185 – ARM EXPRESS-G Graphic 185 of 243

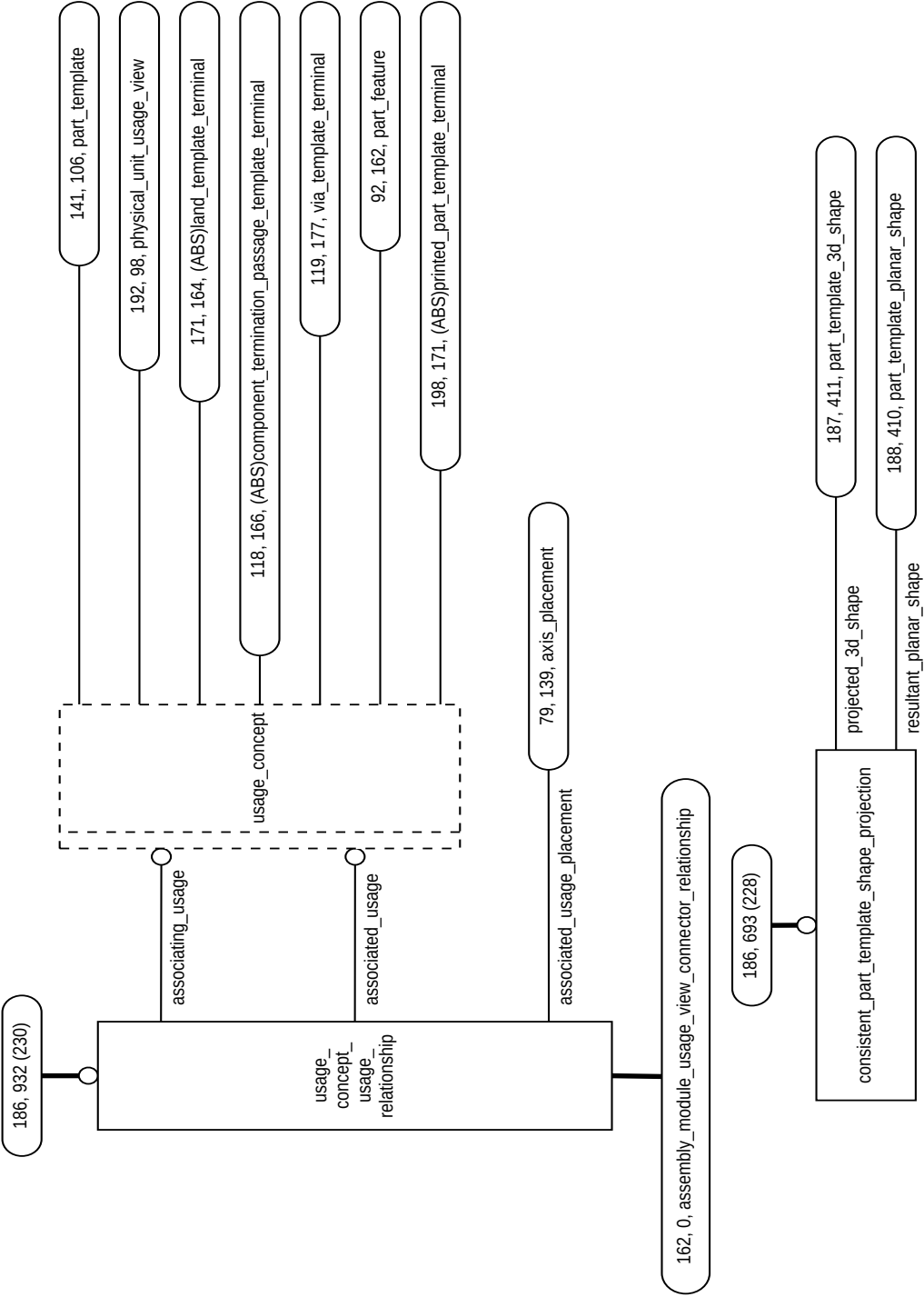


Figure H.186 – ARM EXPRESS-G Graphic 186 of 243

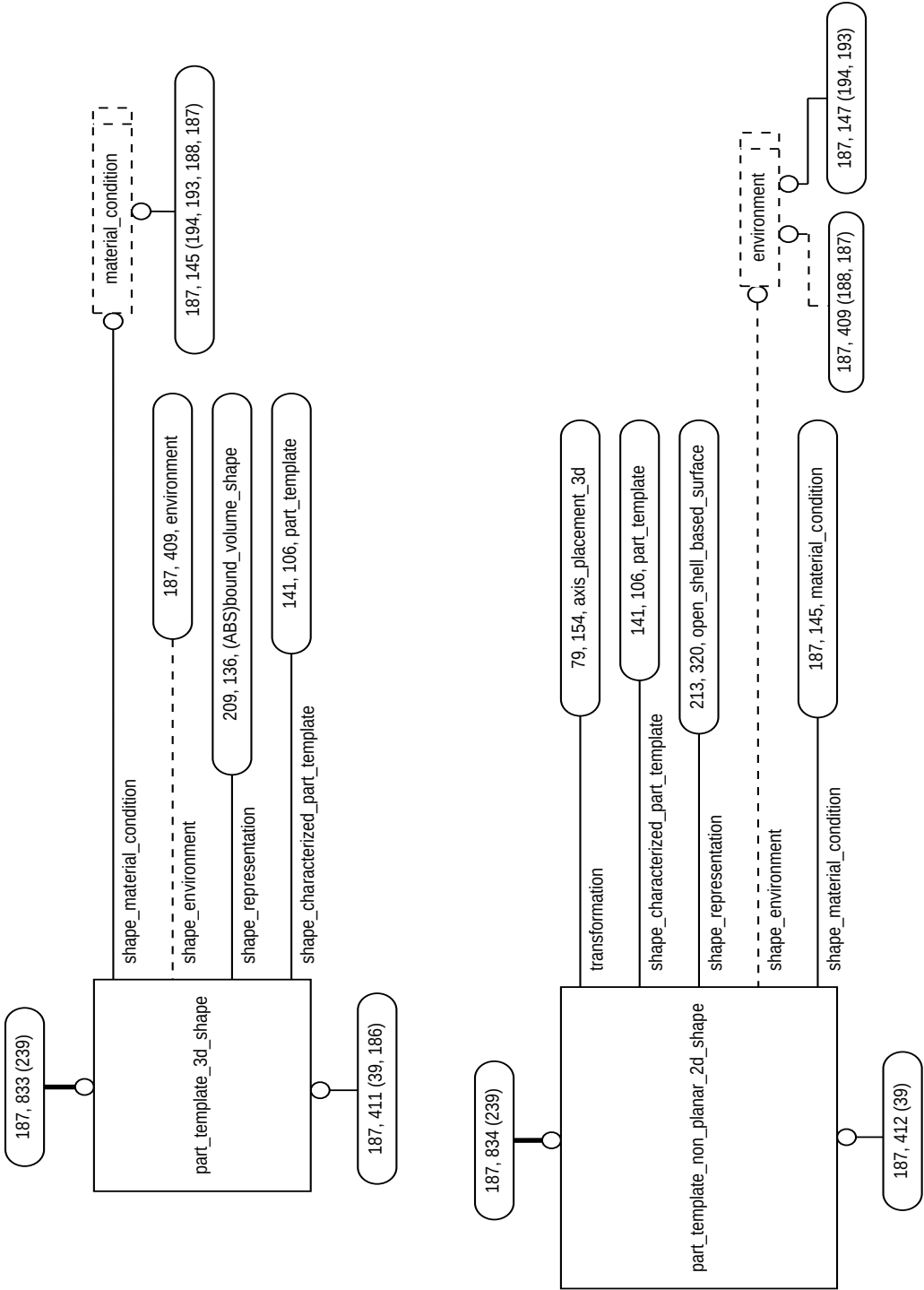


Figure H.187 – ARM EXPRESS-G Graphic 187 of 243

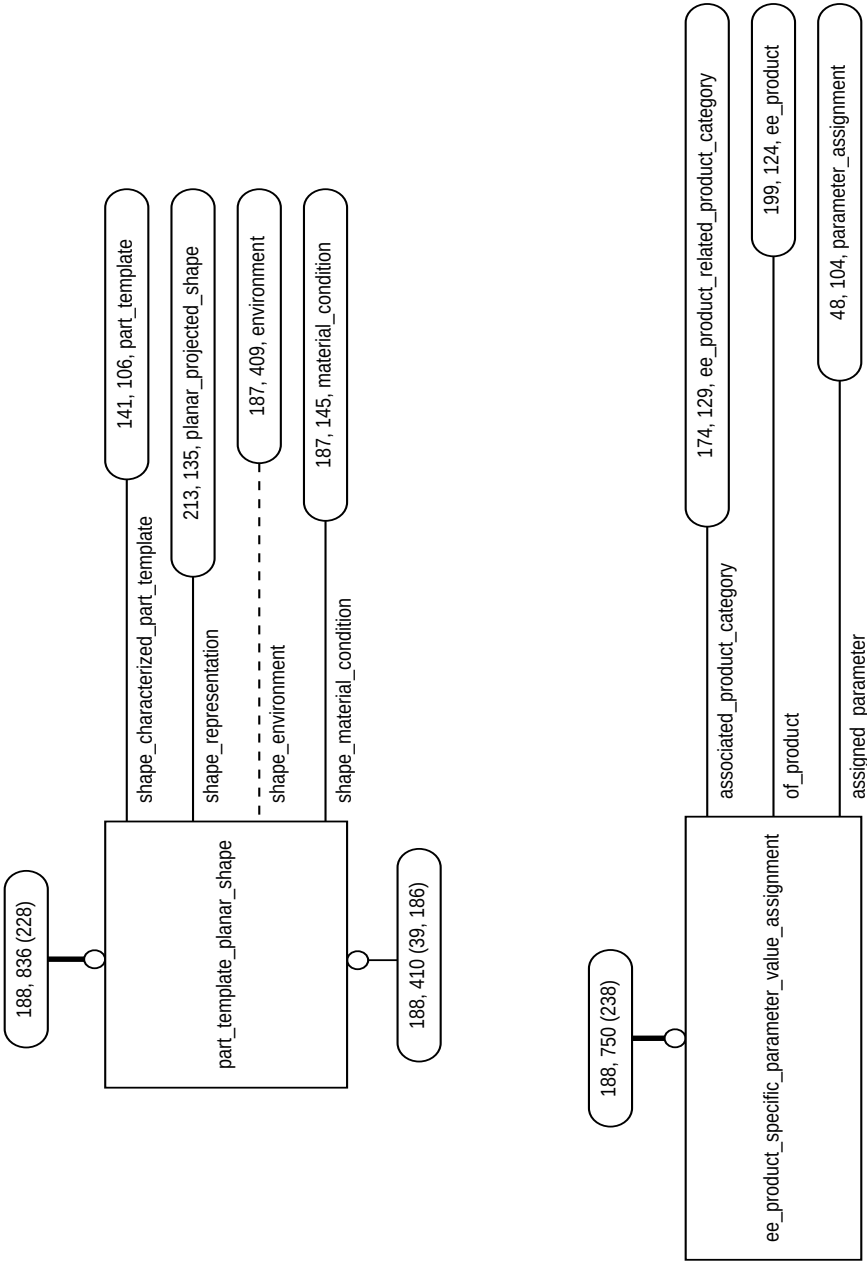


Figure H.188 – ARM EXPRESS-G Graphic 188 of 243

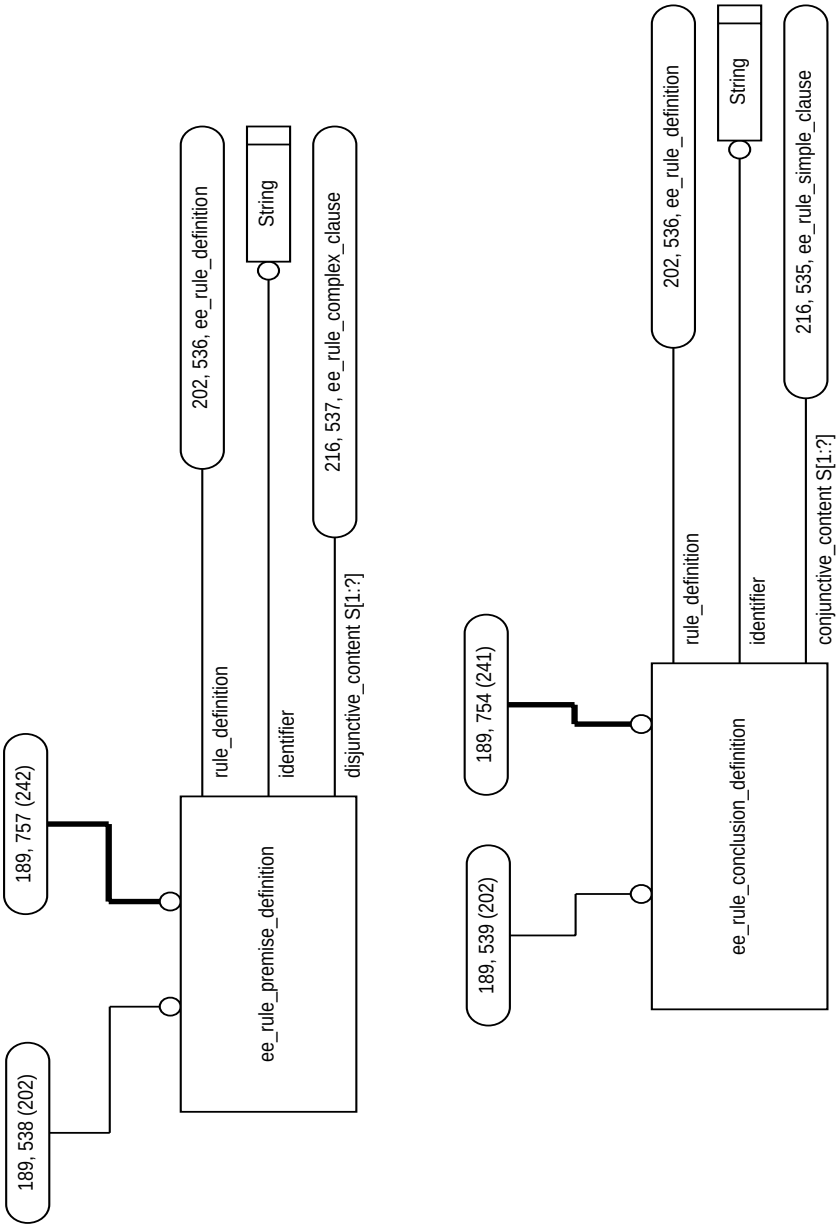


Figure H.189 – ARM EXPRESS-G Graphic 189 of 243

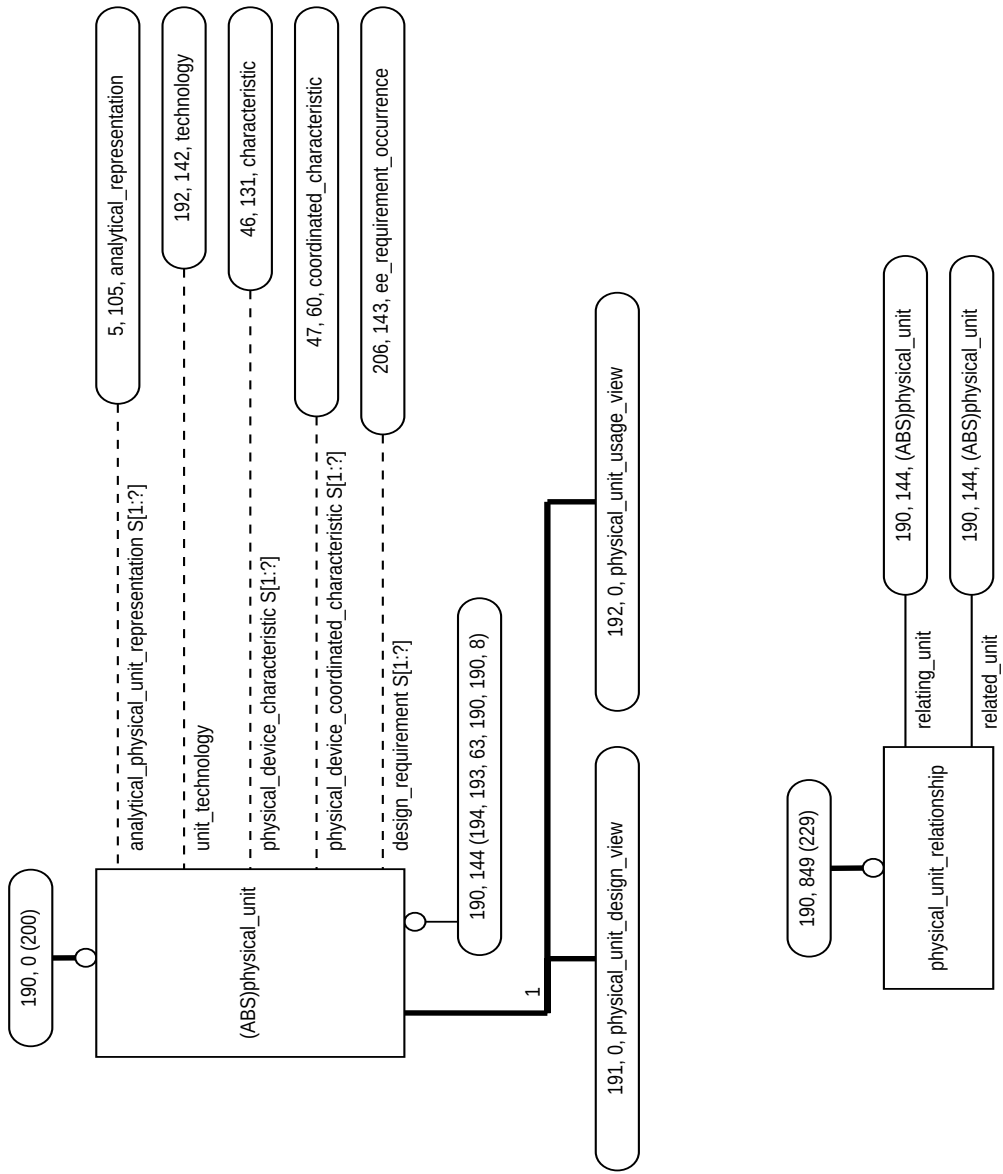


Figure H.190 – ARM EXPRESS-G Graphic 190 of 243

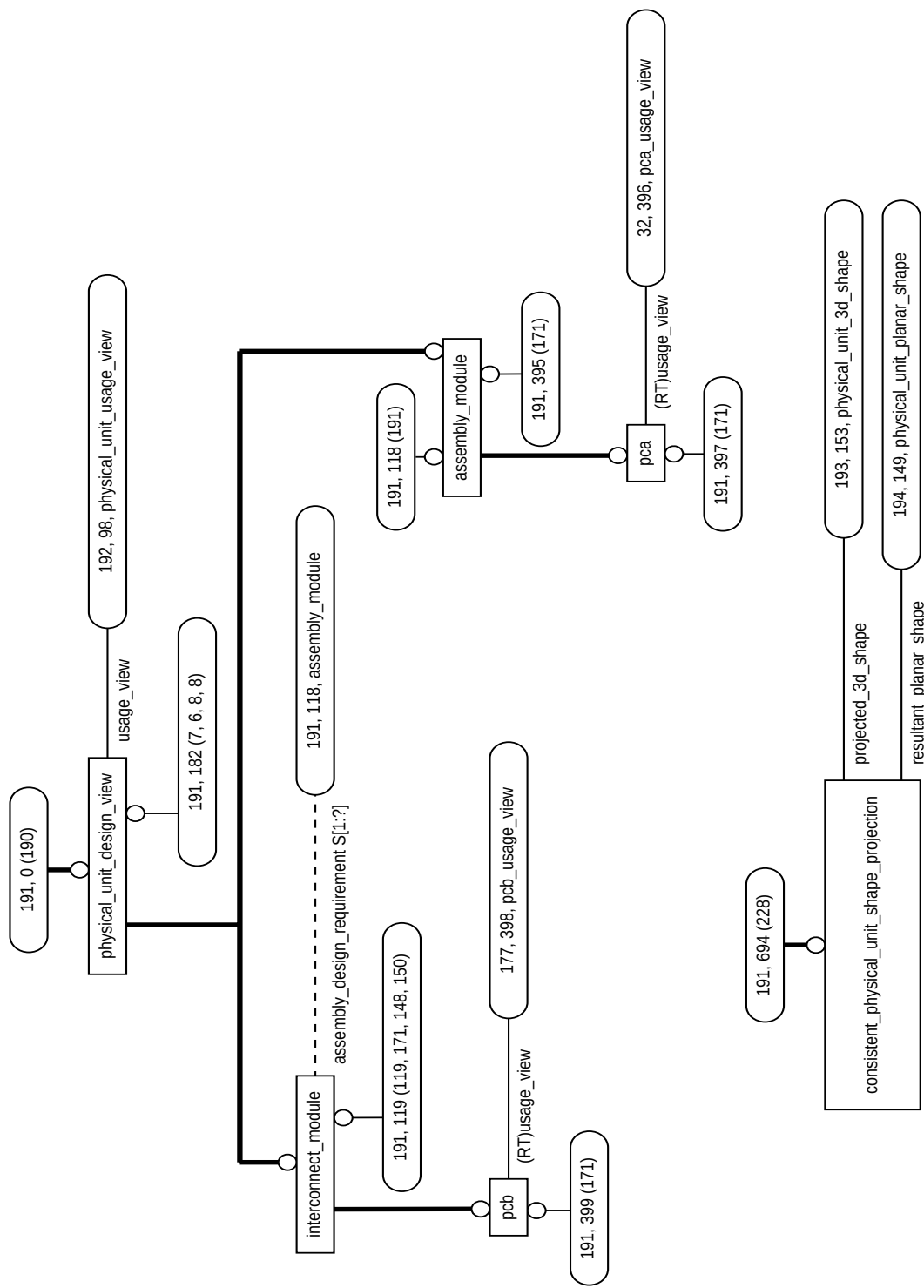


Figure H.191 – ARM EXPRESS-G Graphic 191 of 243

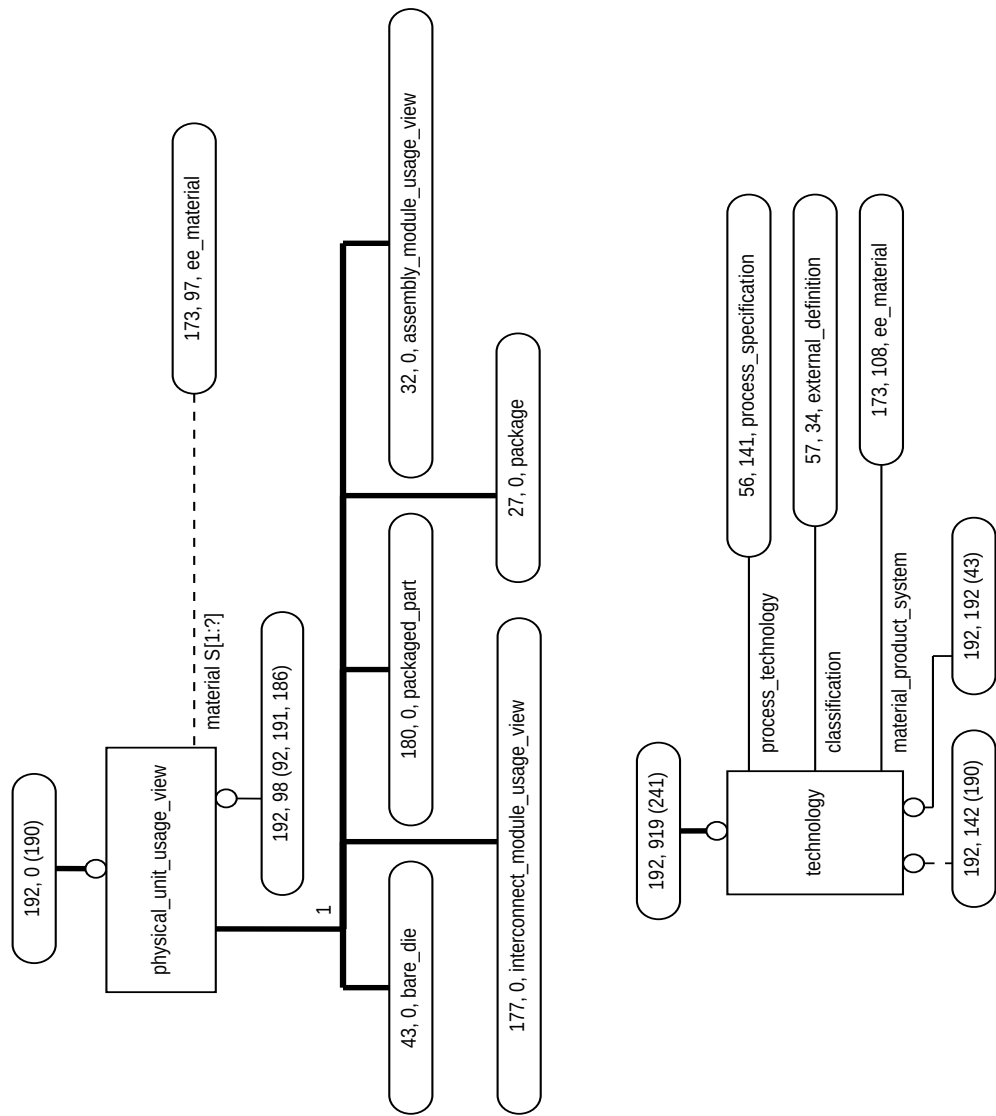


Figure H.192 – ARM EXPRESS-G Graphic 192 of 243

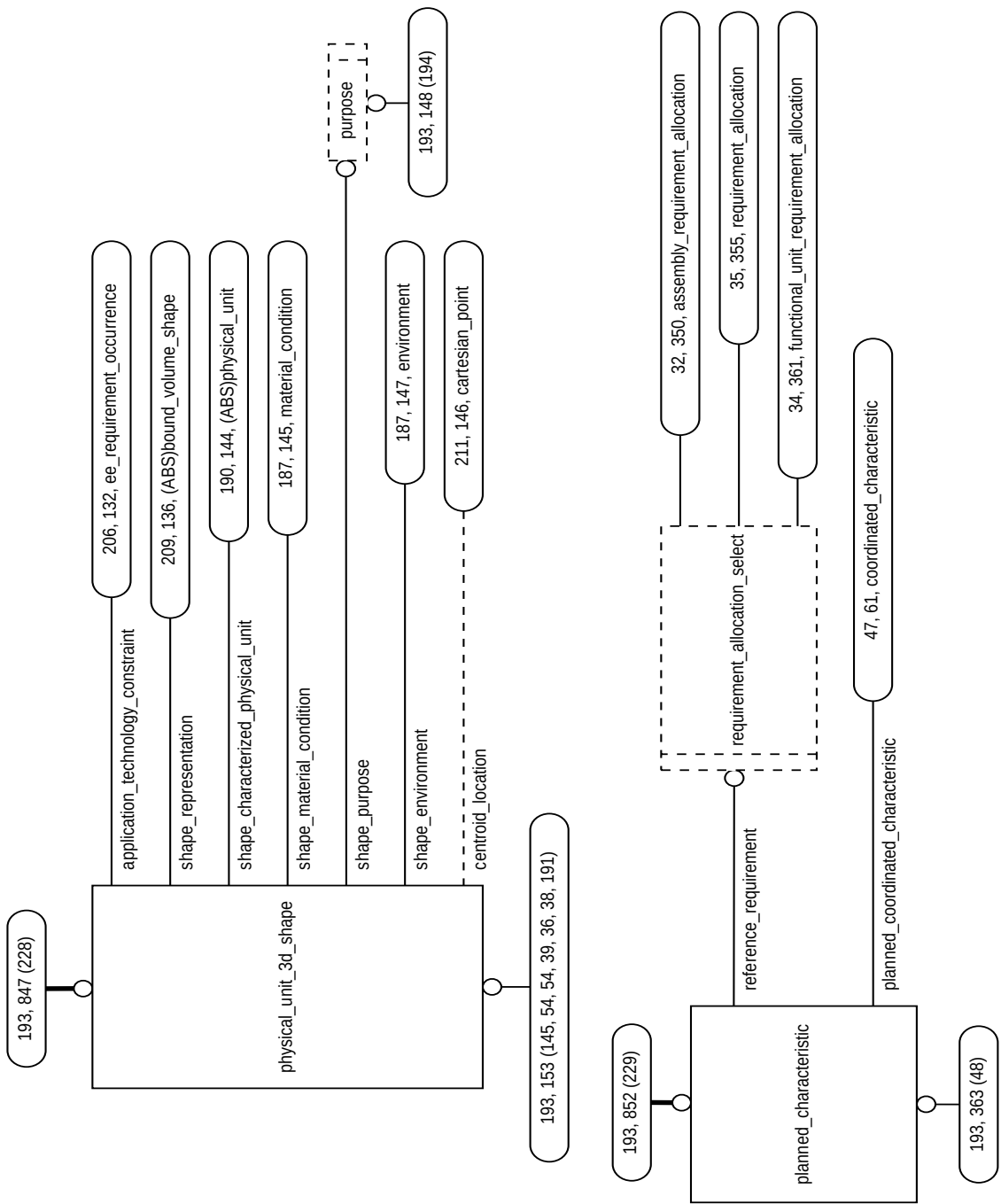


Figure H.193 – ARM EXPRESS-G Graphic 193 of 243

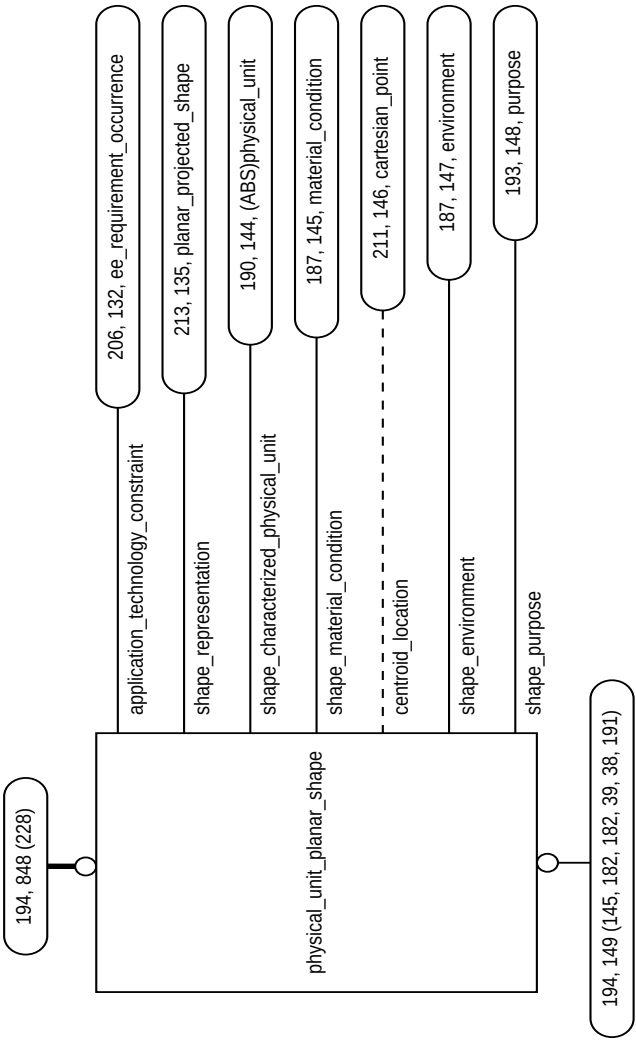


Figure H.194 – ARM EXPRESS-G Graphic 194 of 243

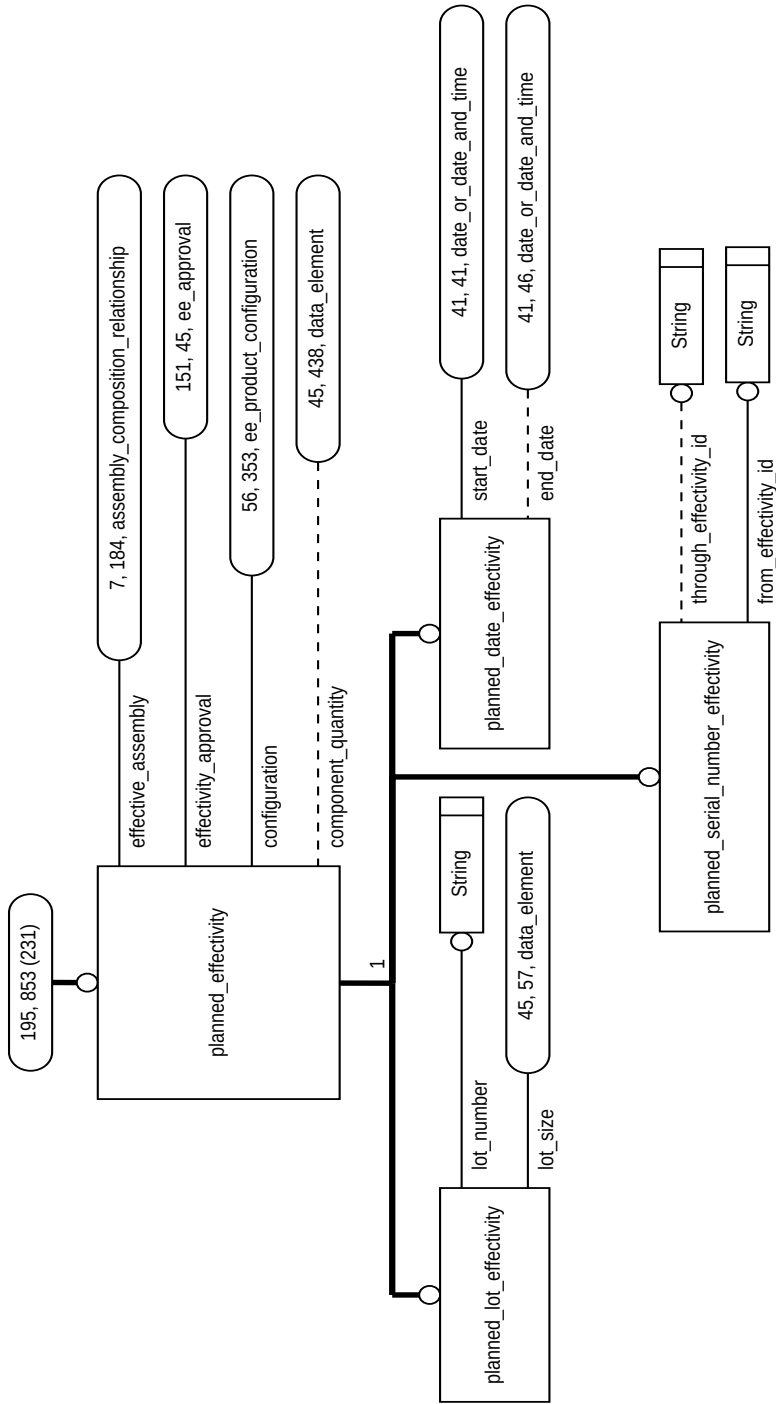


Figure H.195 – ARM EXPRESS-G Graphic 195 of 243

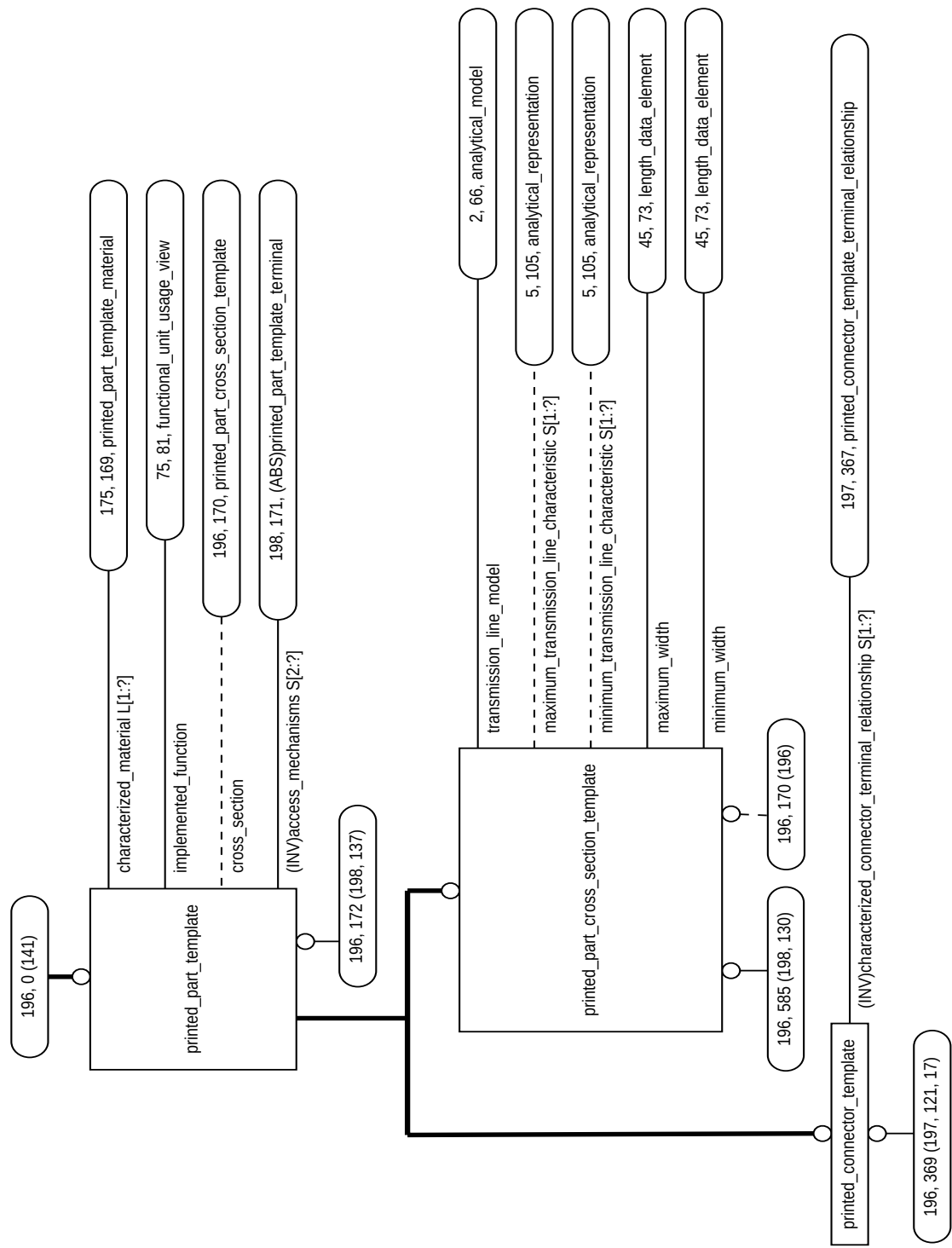


Figure H.196 – ARM EXPRESS-G Graphic 196 of 243

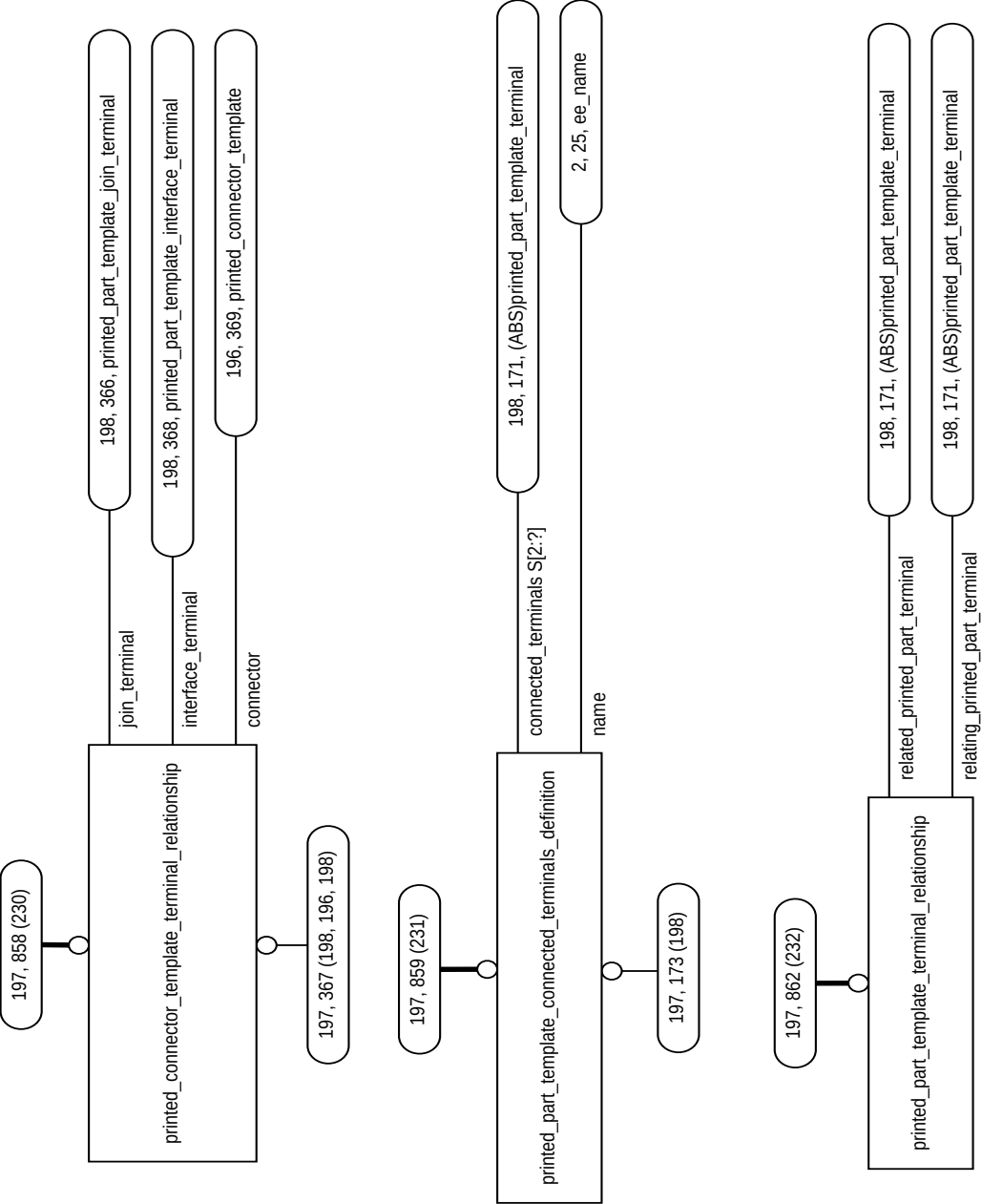


Figure H.197 – ARM EXPRESS-G Graphic 197 of 243

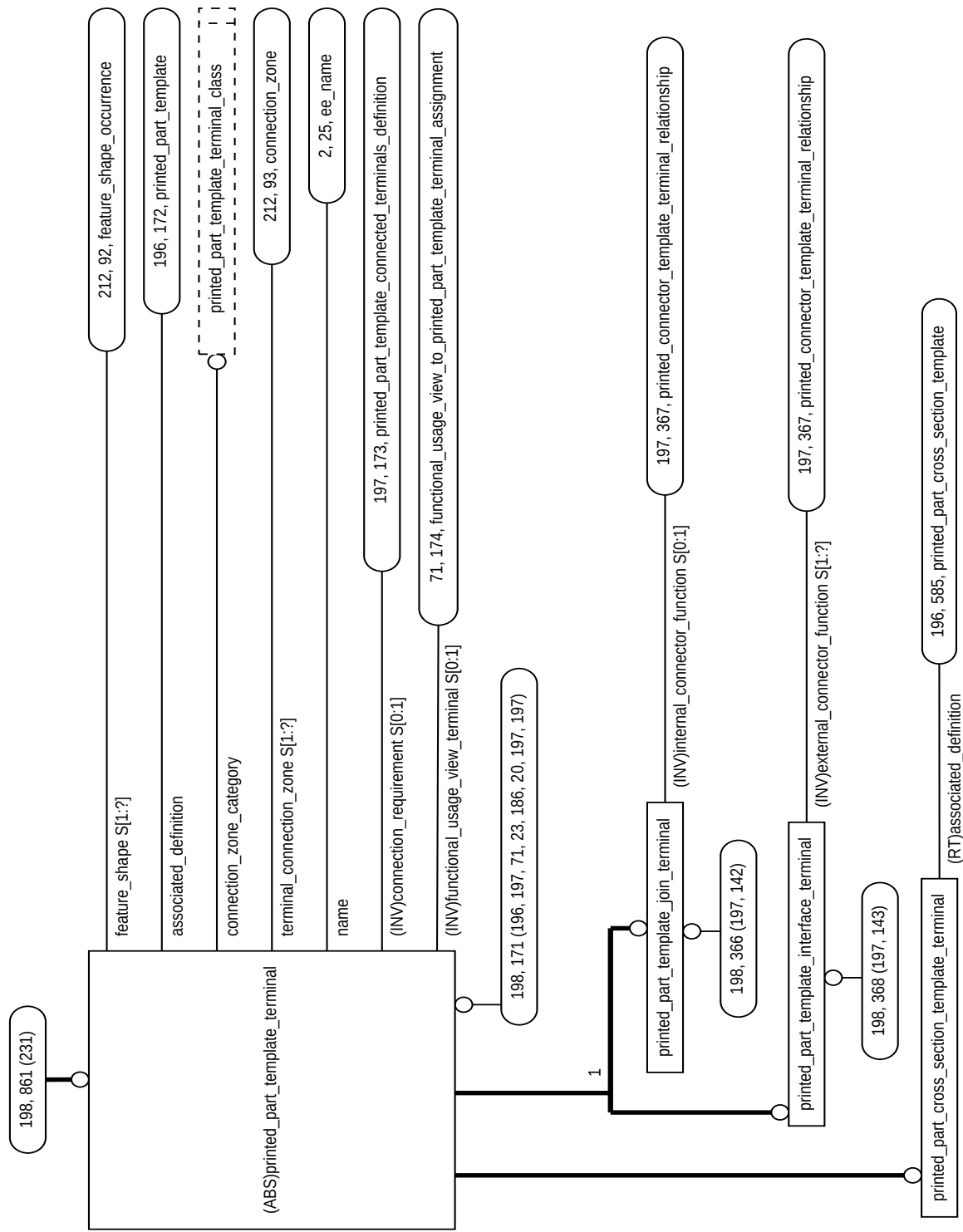


Figure H.198 – ARM EXPRESS-G Graphic 198 of 243

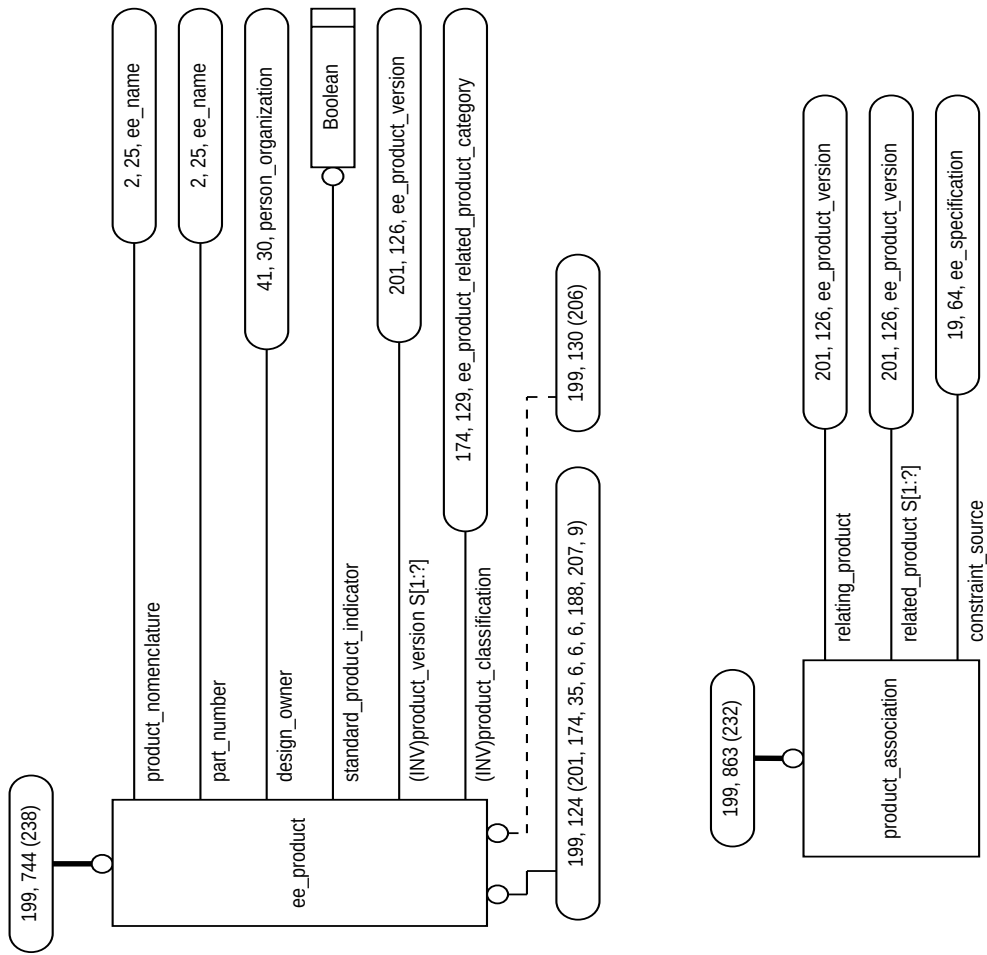


Figure H.199 – ARM EXPRESS-G Graphic 199 of 243

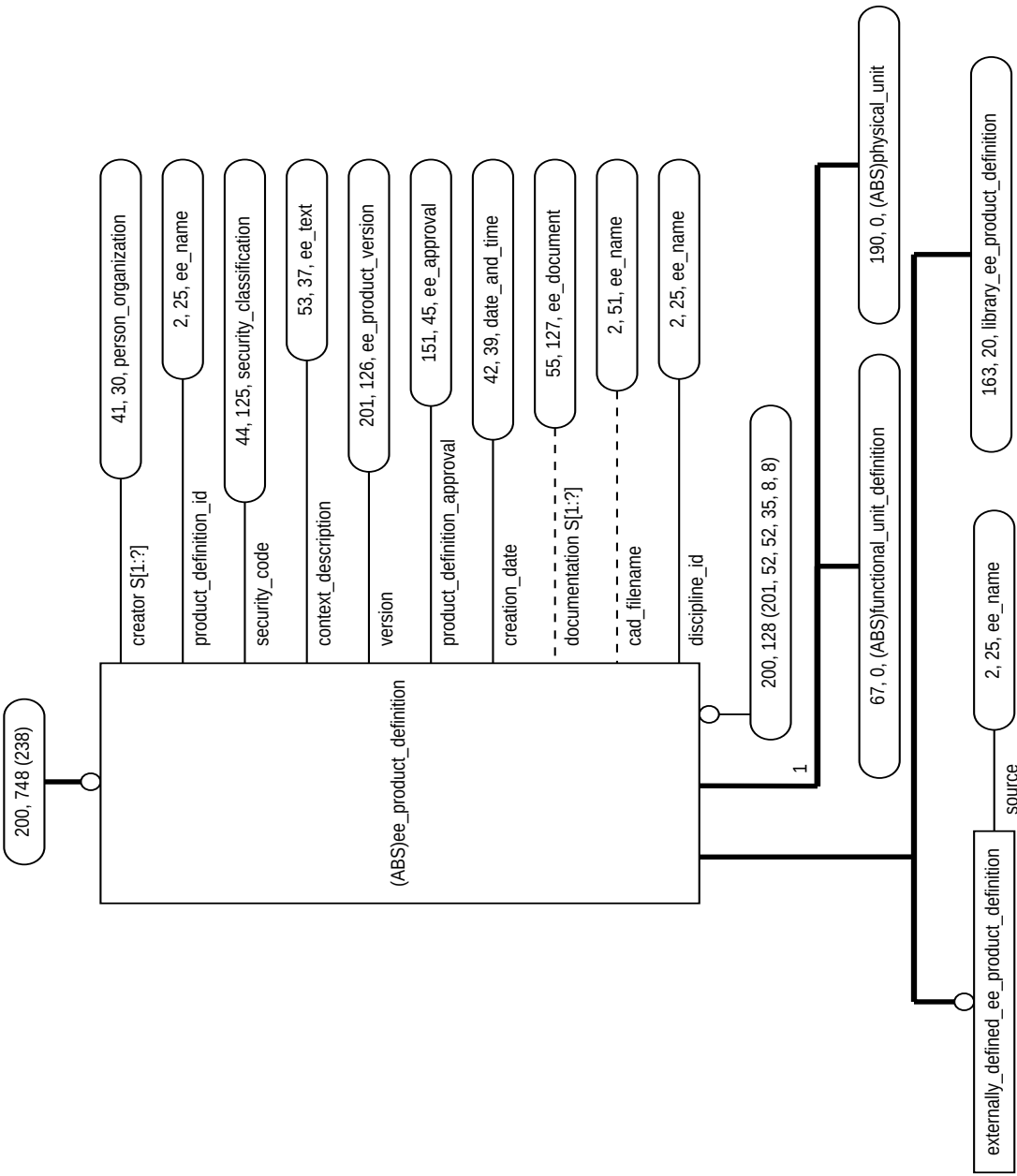


Figure H.200 – ARM EXPRESS-G Graphic 200 of 243

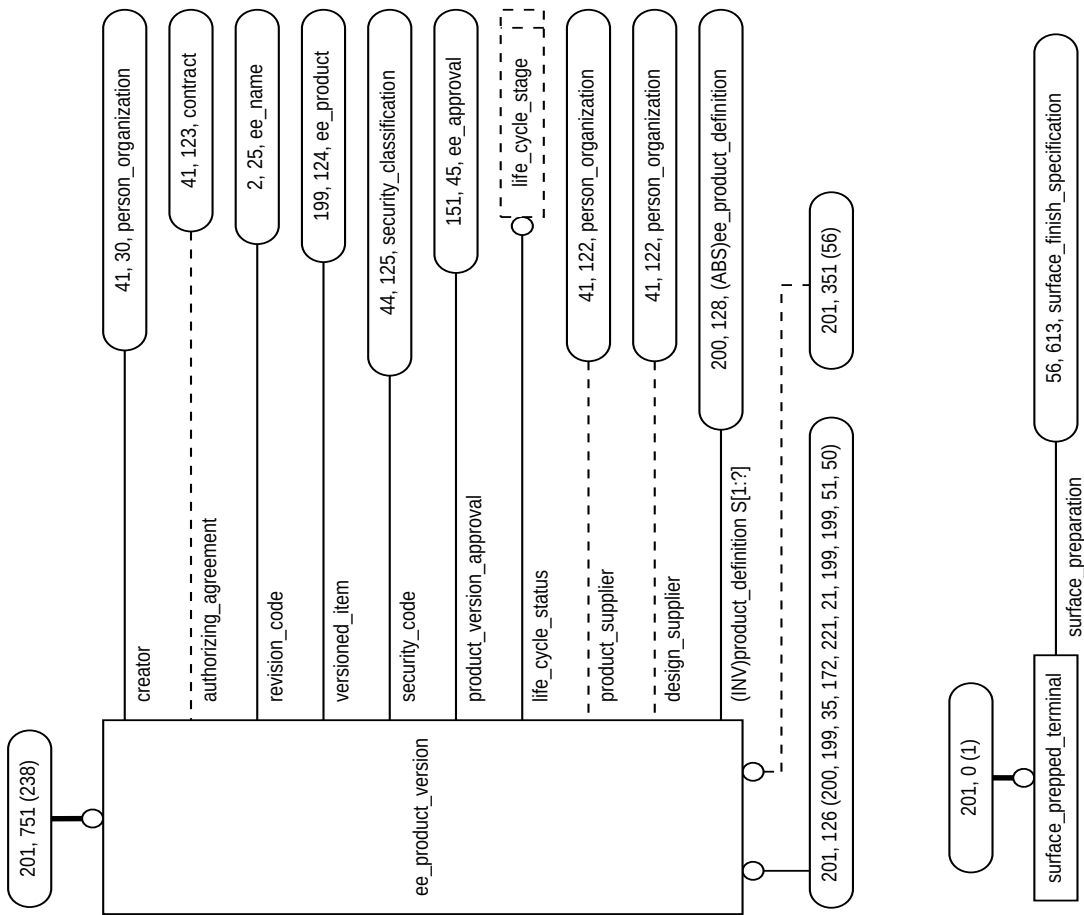


Figure H.201 – ARM EXPRESS-G Graphic 201 of 243

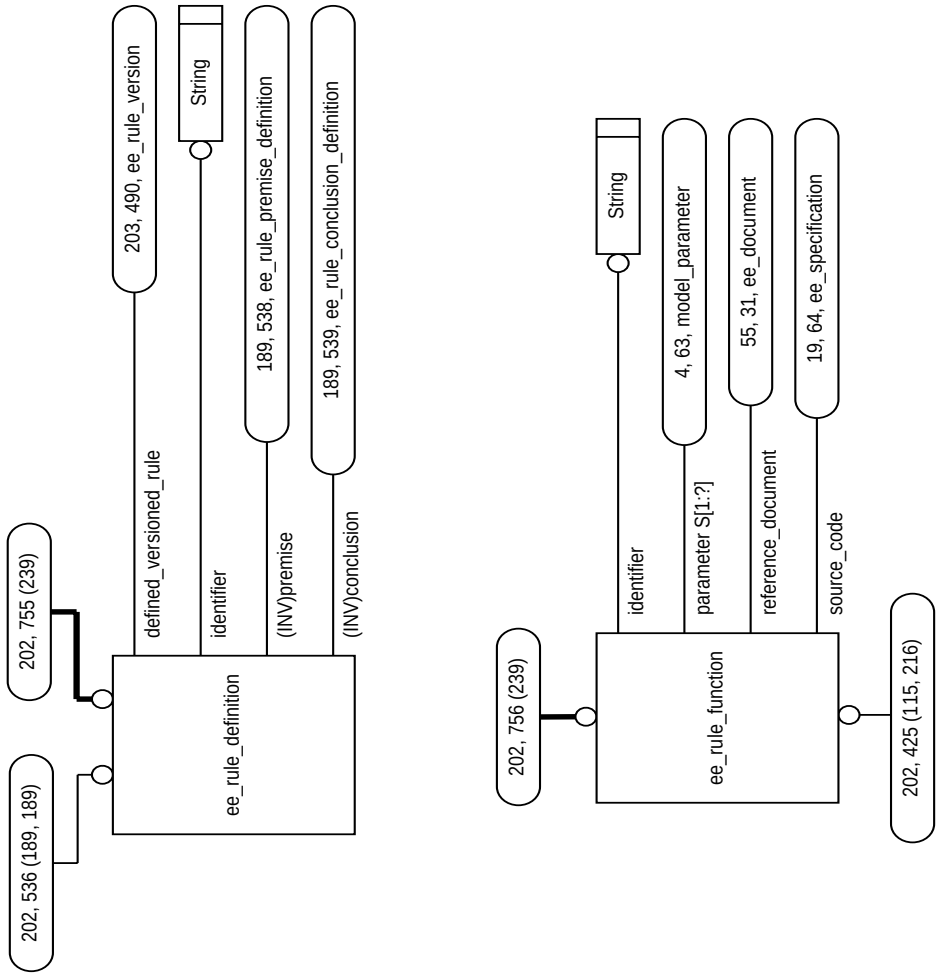


Figure H.202 – ARM EXPRESS-G Graphic 202 of 243

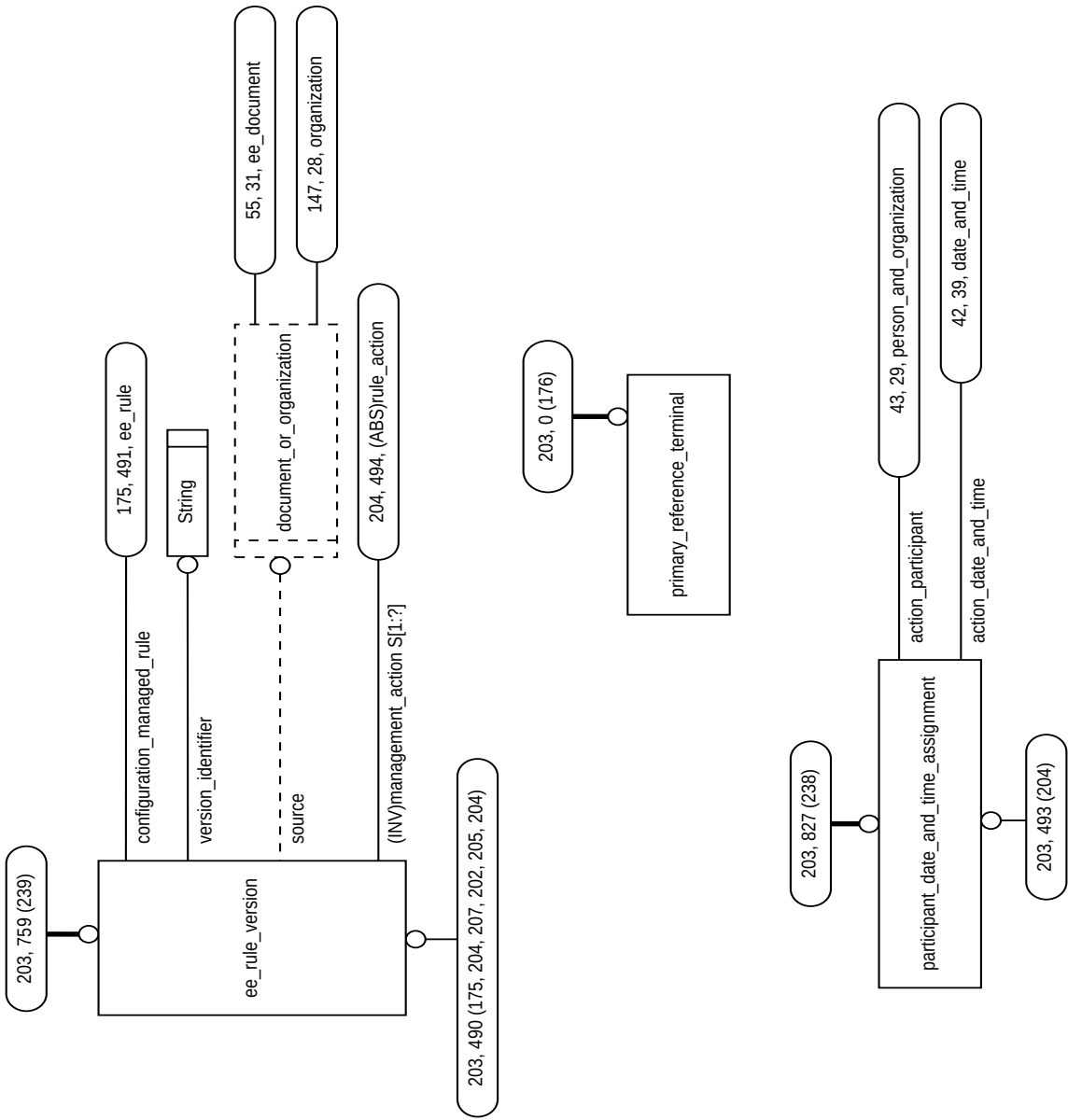


Figure H.203 – ARM EXPRESS-G Graphic 203 of 243

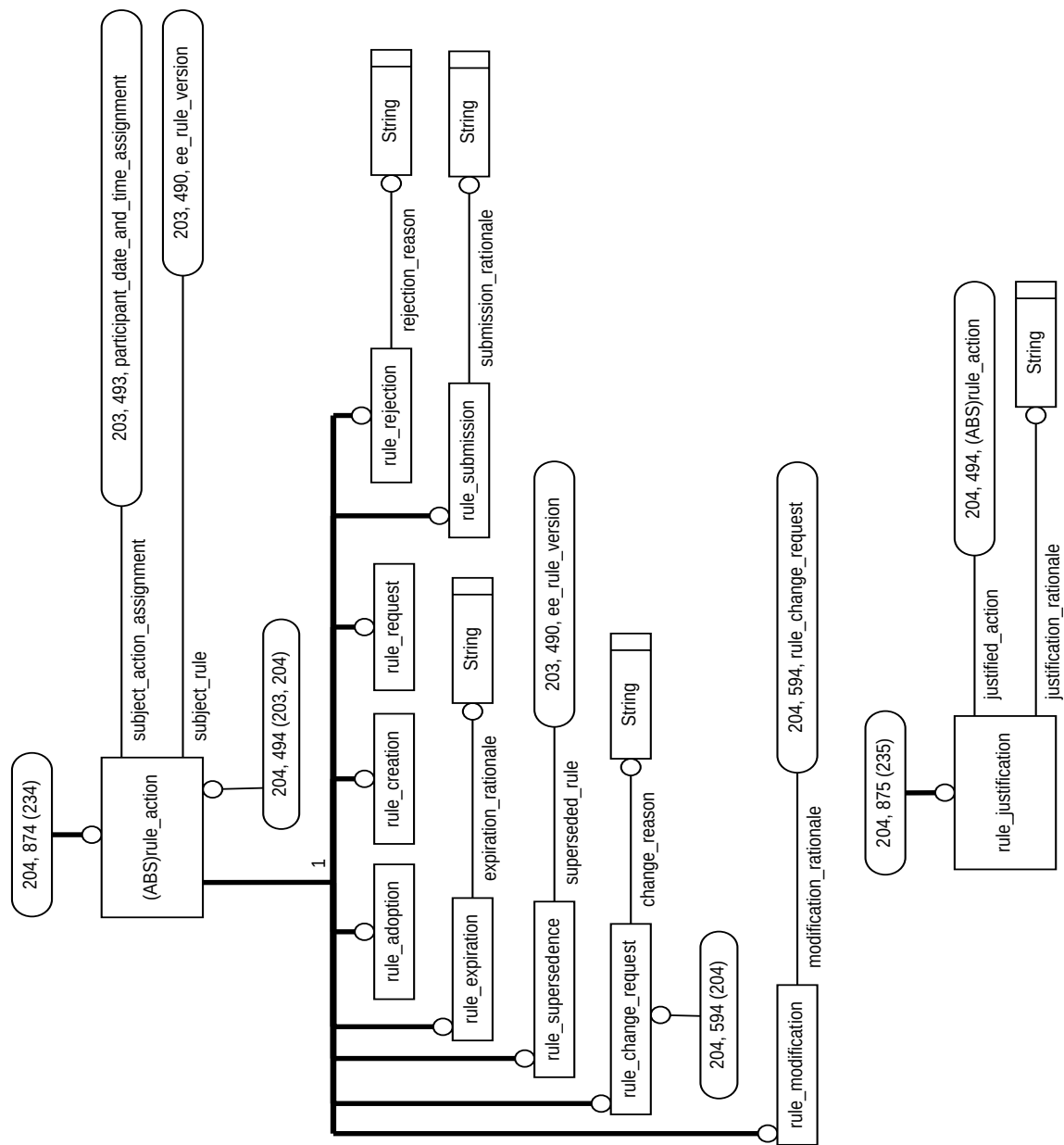


Figure H.204 – ARM EXPRESS-G Graphic 204 of 243

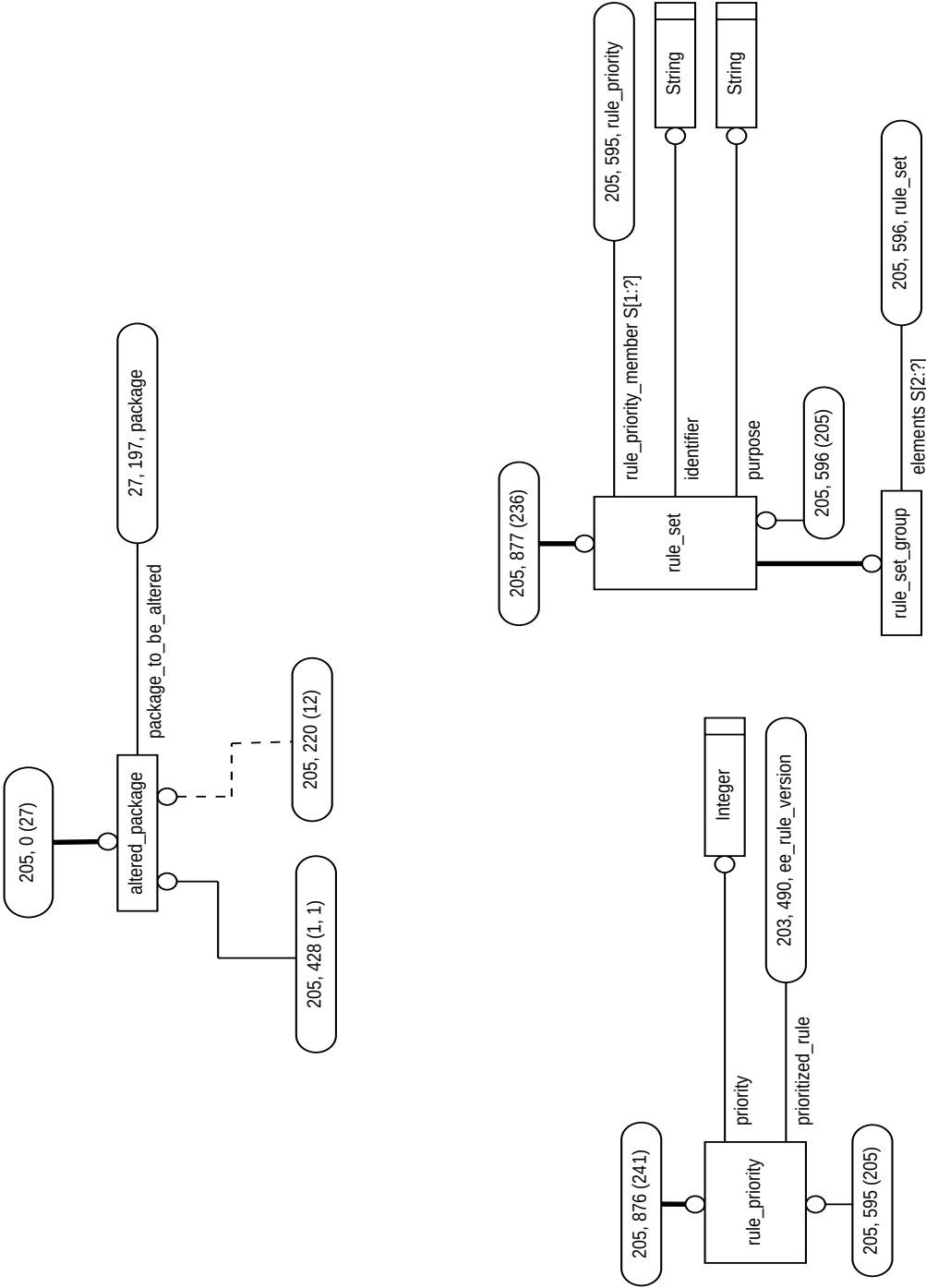


Figure H.205 – ARM EXPRESS-G Graphic 205 of 243

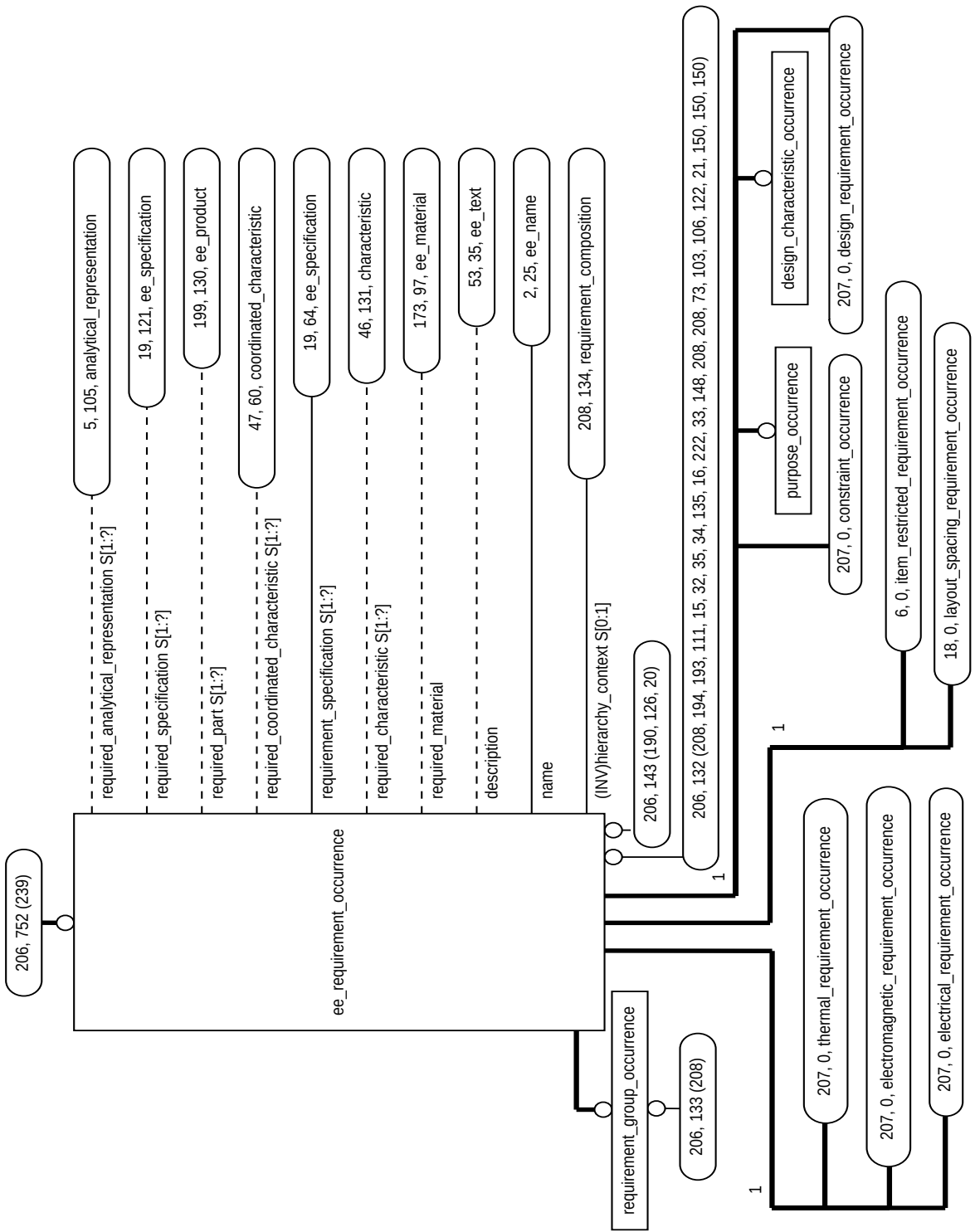


Figure H.206 – ARM EXPRESS-G Graphic 206 of 243

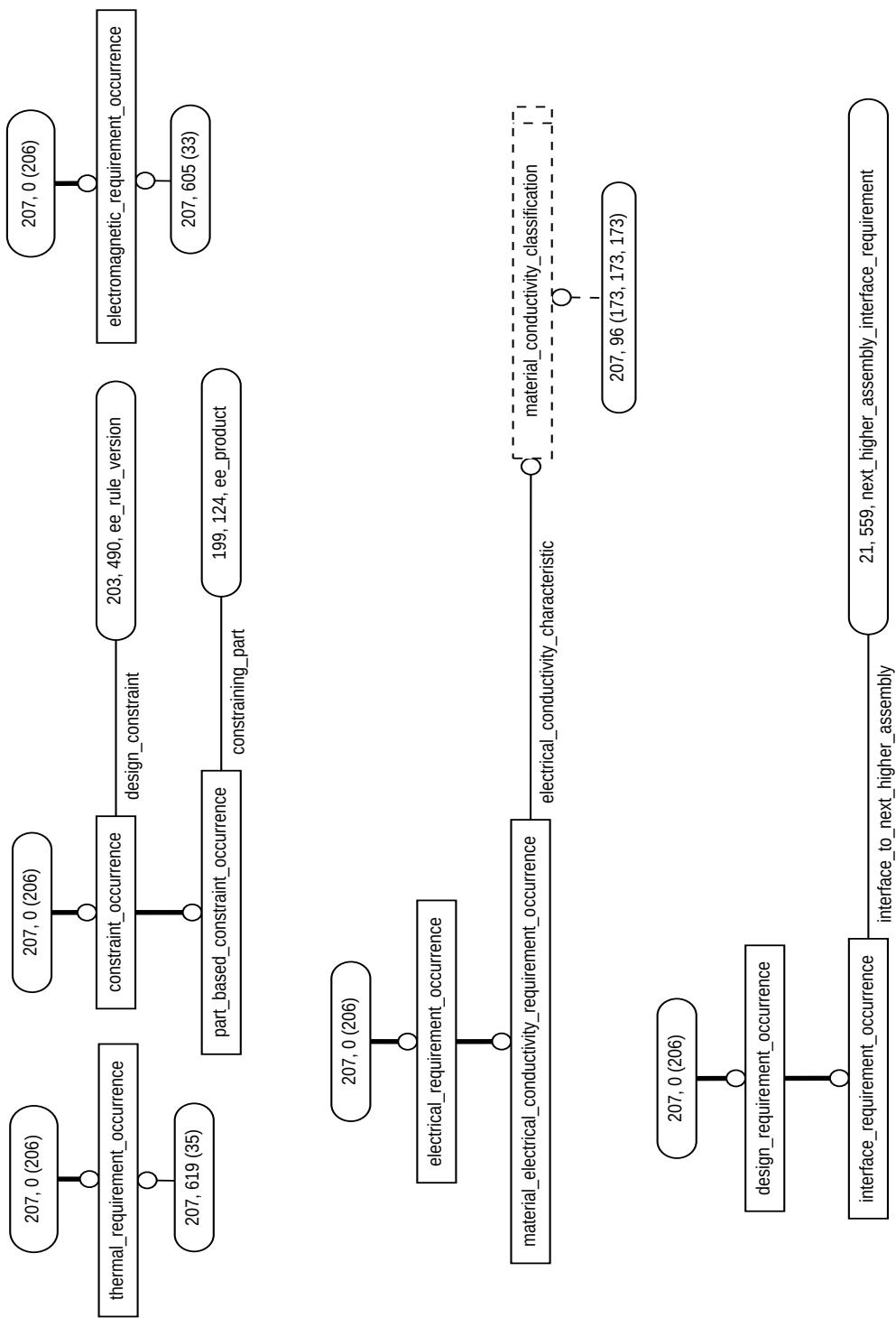


Figure H.207 – ARM EXPRESS-G Graphic 207 of 243

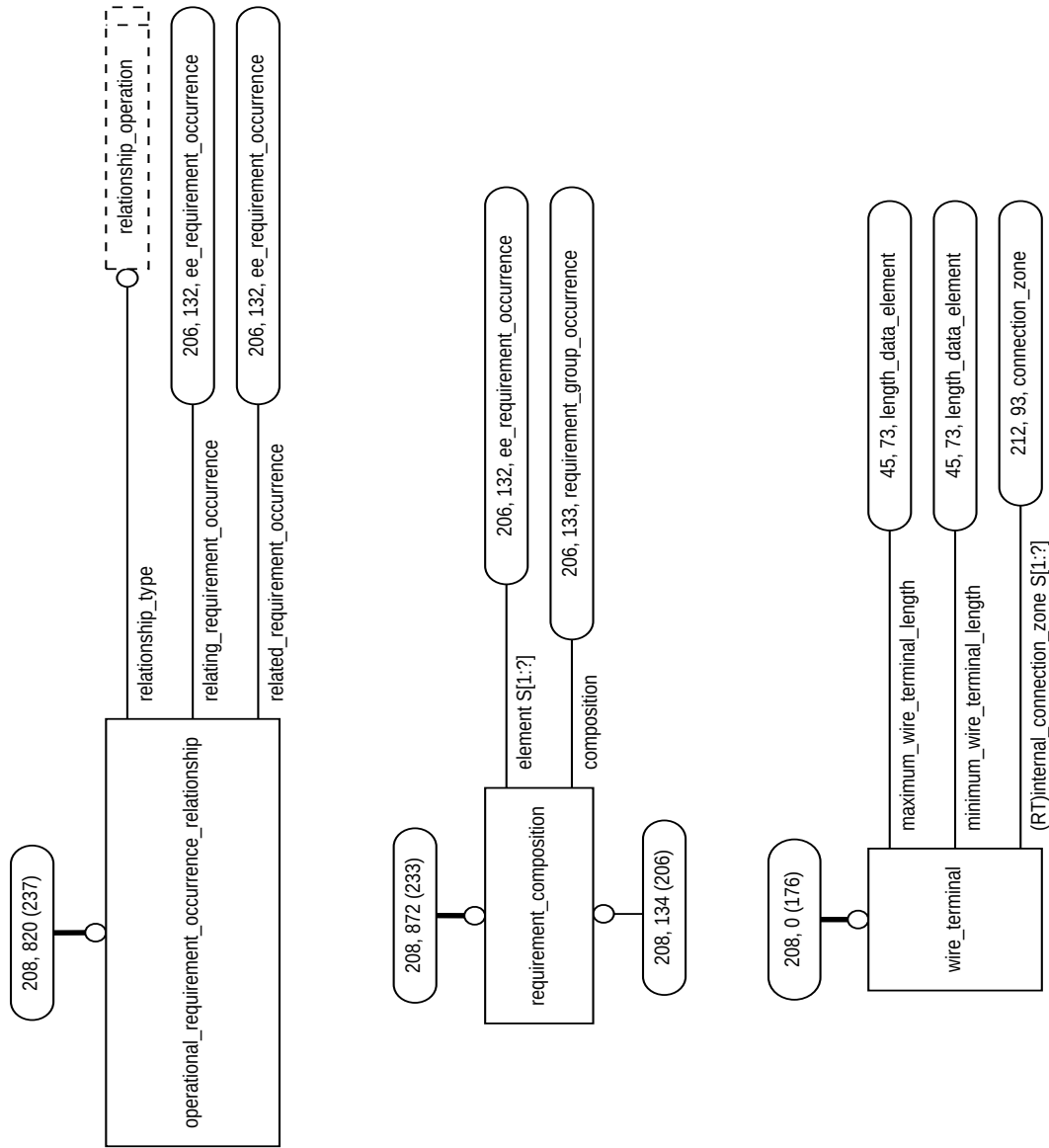


Figure H.208 – ARM EXPRESS-G Graphic 208 of 243

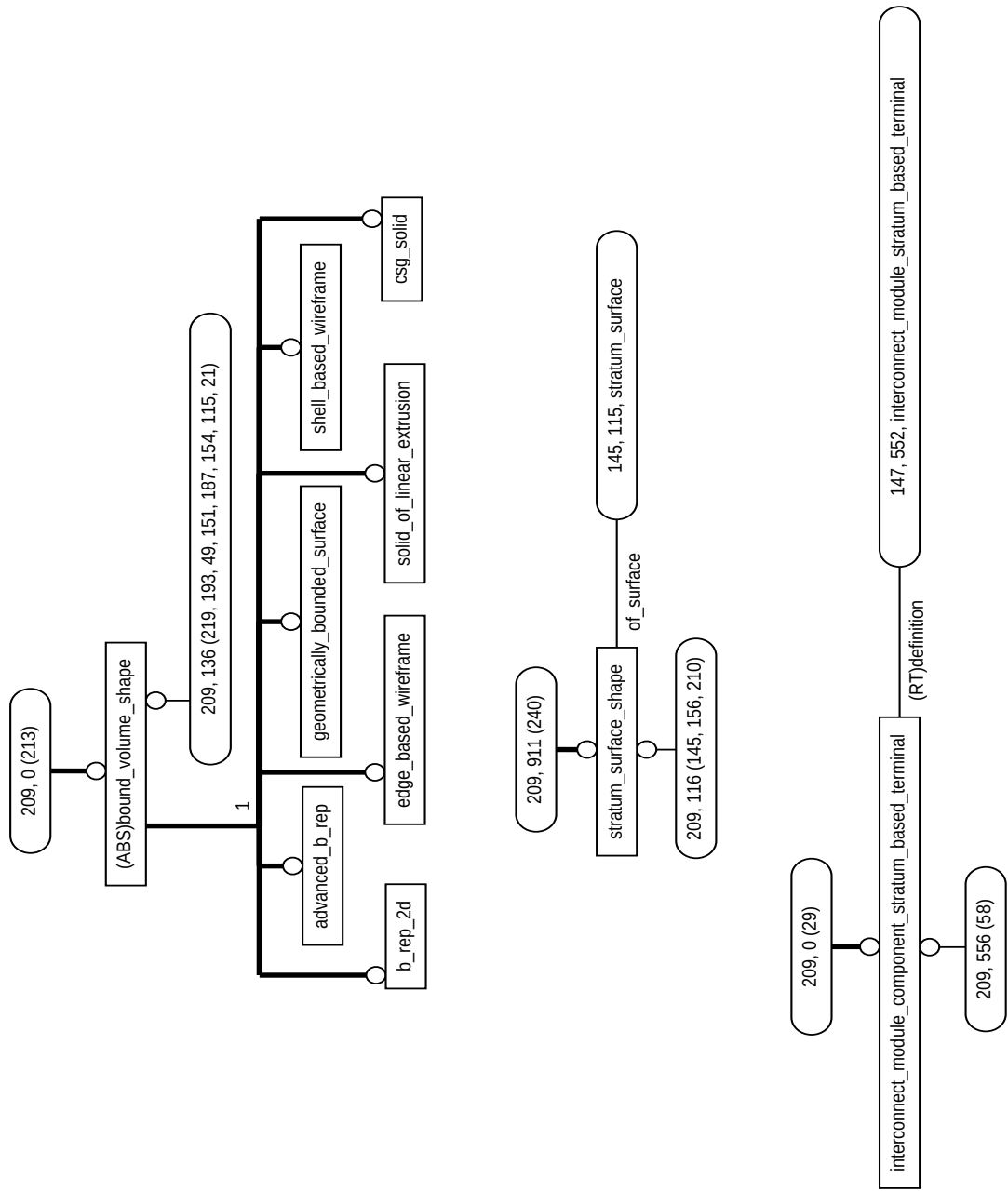


Figure H.209 – ARM EXPRESS-G Graphic 209 of 243

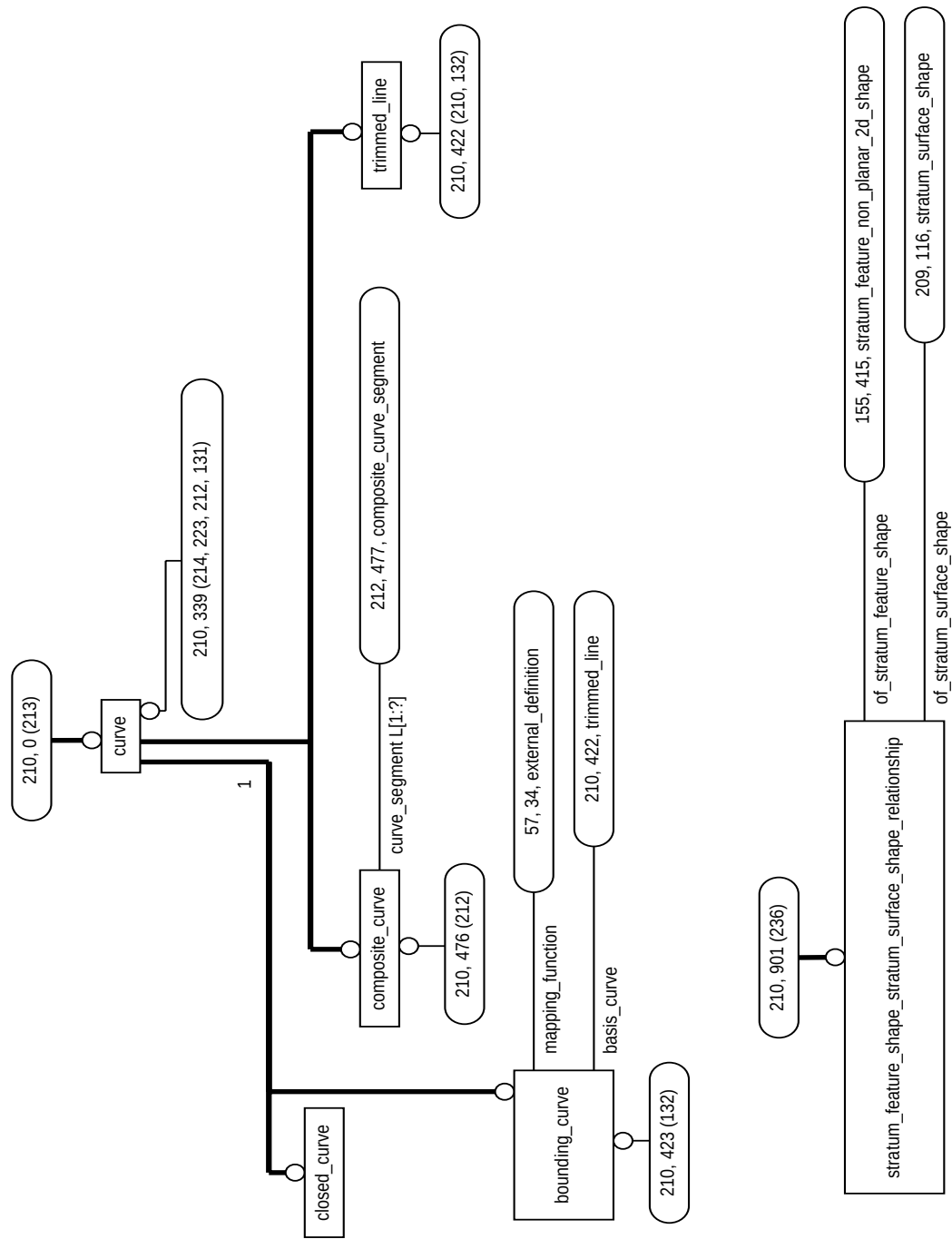


Figure H.210 – ARM EXPRESS-G Graphic 210 of 243

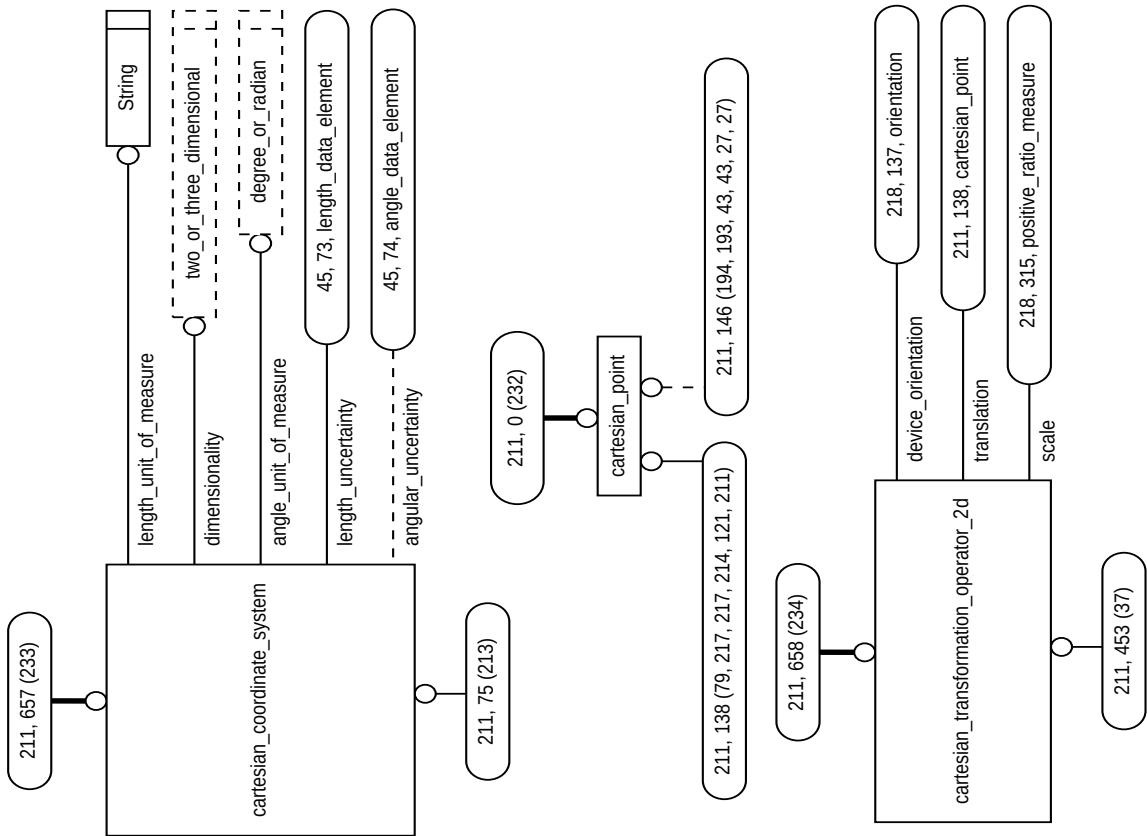


Figure H.211 – ARM EXPRESS-G Graphic 211 of 243

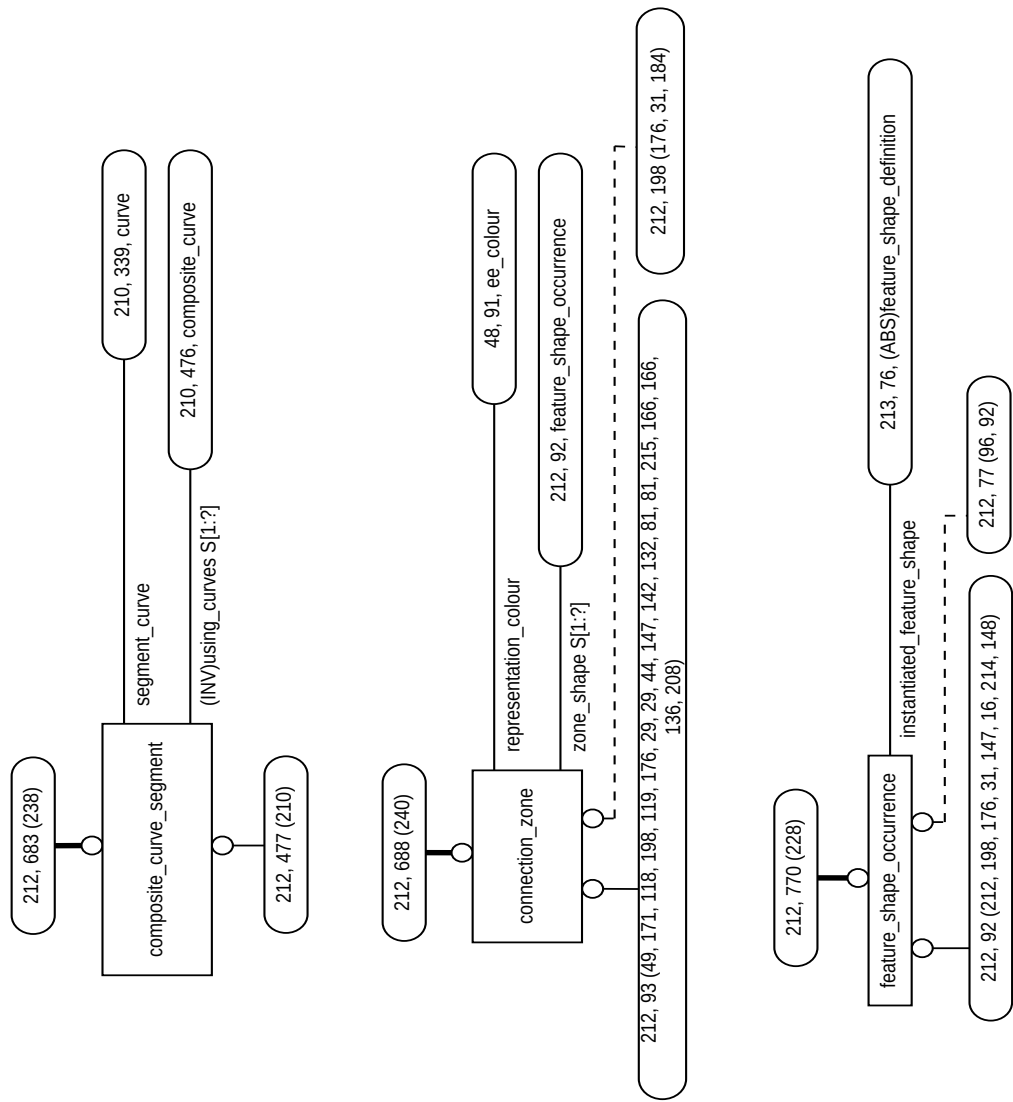


Figure H.212 – ARM EXPRESS-G Graphic 212 of 243

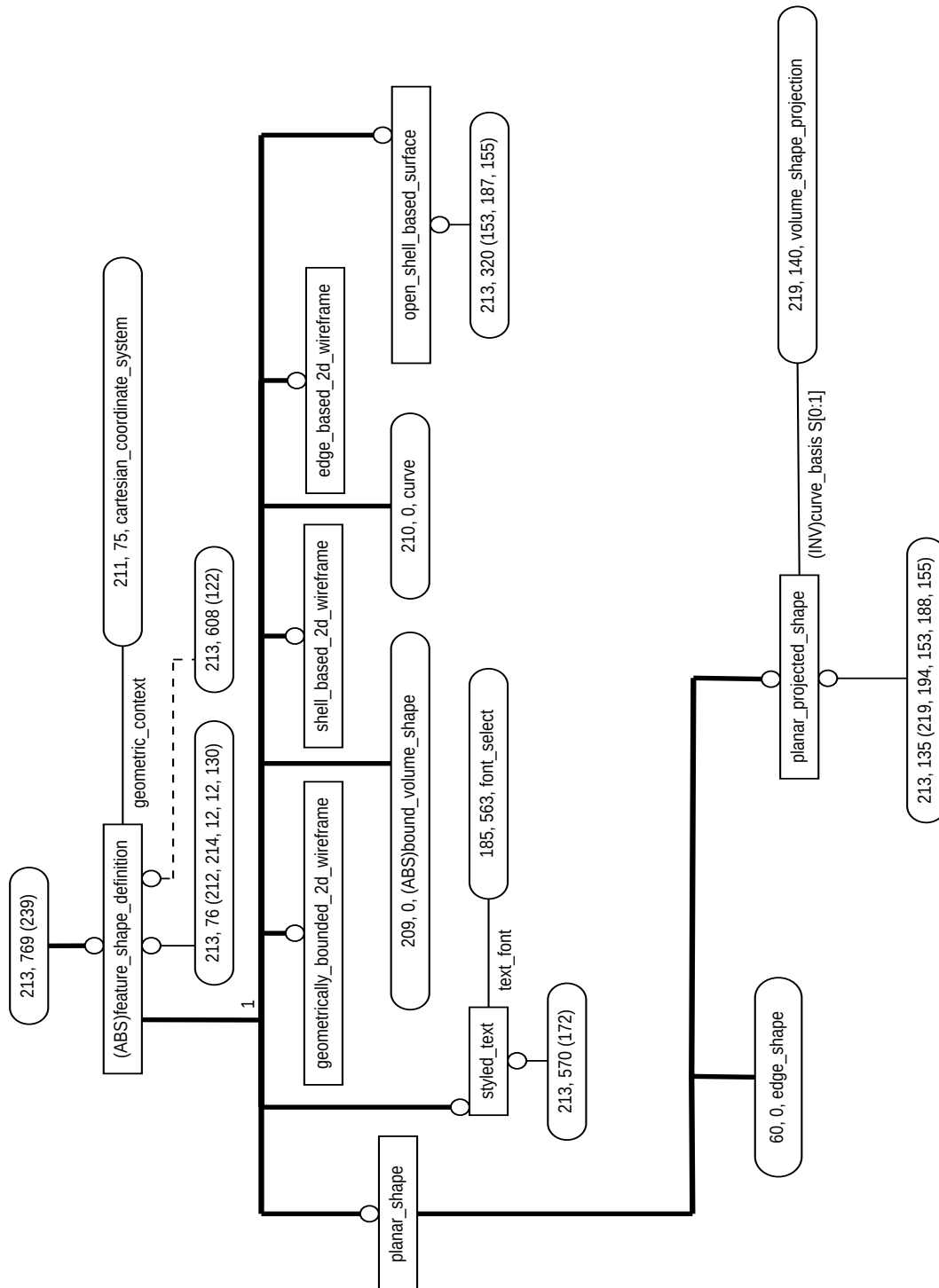


Figure H.213 – ARM EXPRESS-G Graphic 213 of 243

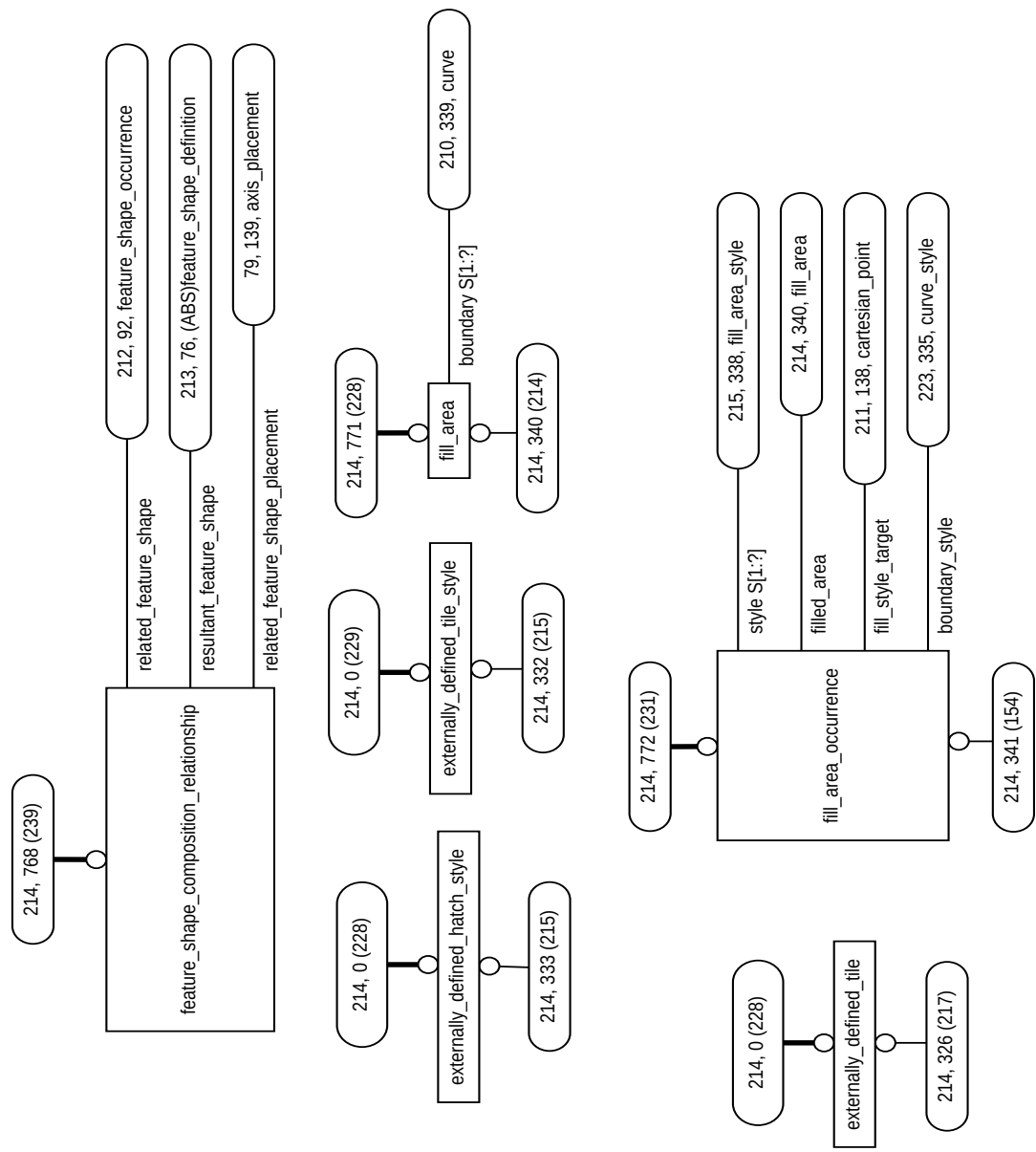


Figure H.214 – ARM EXPRESS-G Graphic 214 of 243

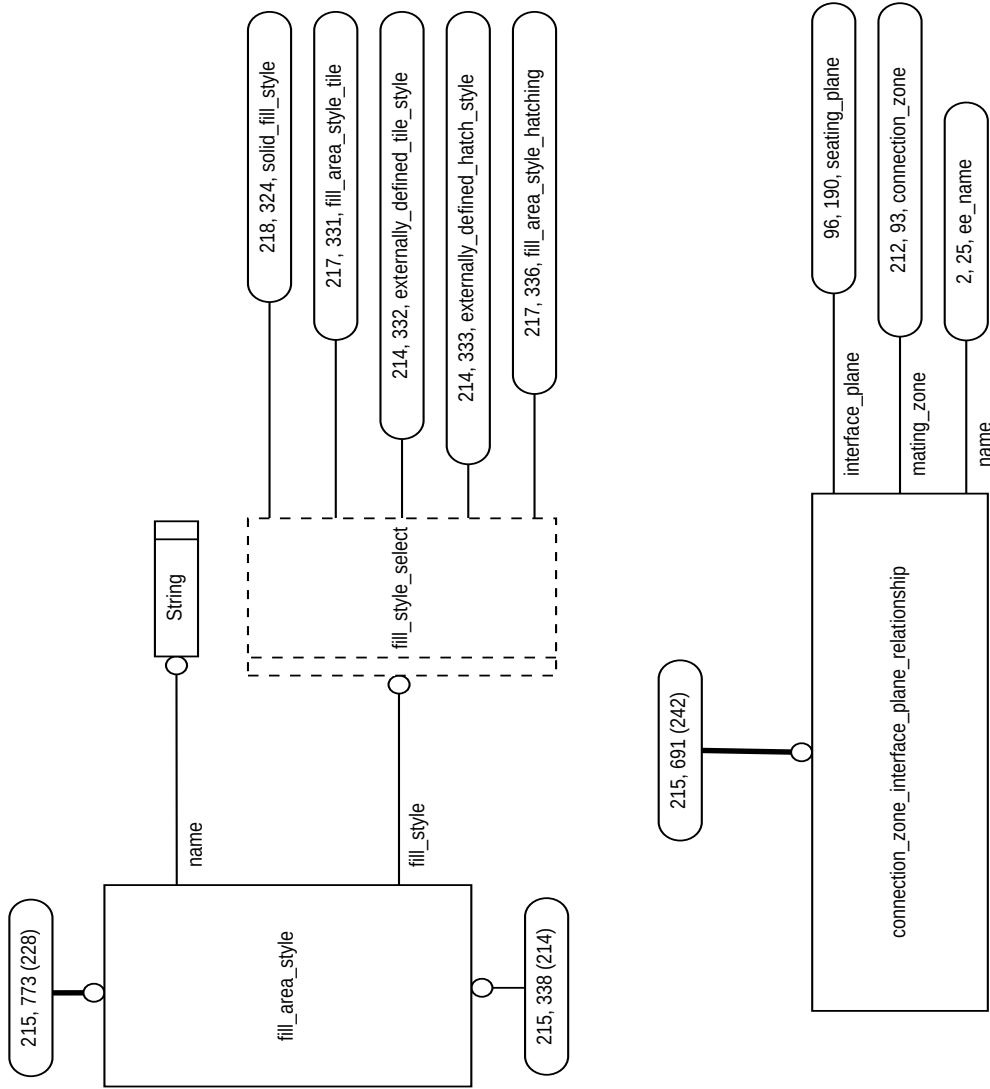


Figure H.215 – ARM EXPRESS-G Graphic 215 of 243

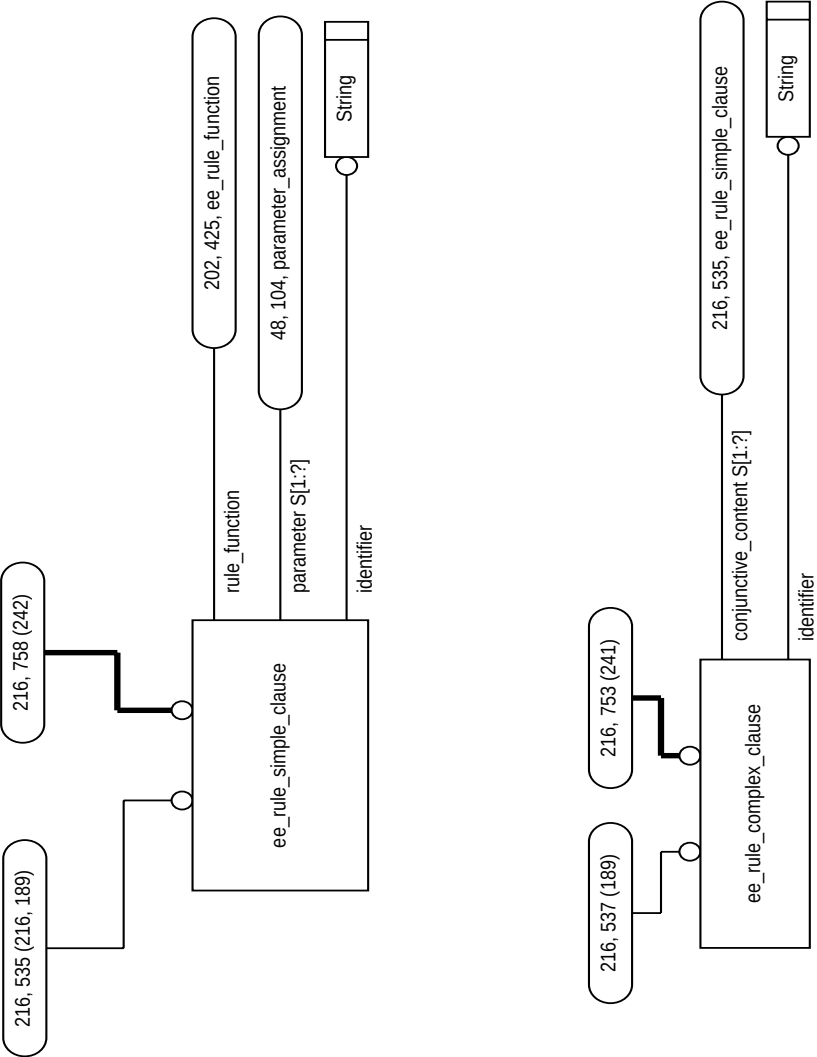


Figure H.216 – ARM EXPRESS-G Graphic 216 of 243

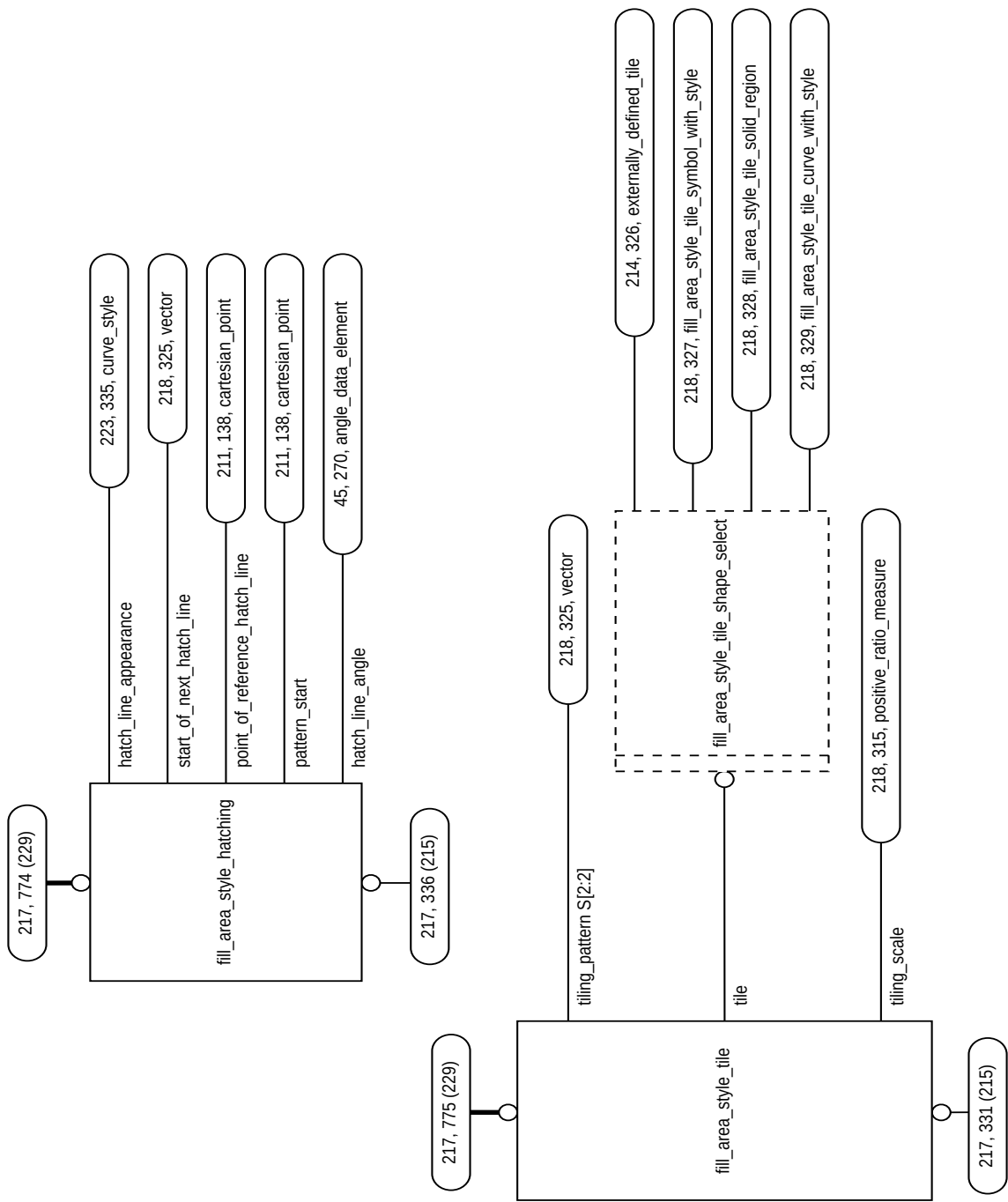


Figure H.217 – ARM EXPRESS-G Graphic 217 of 243

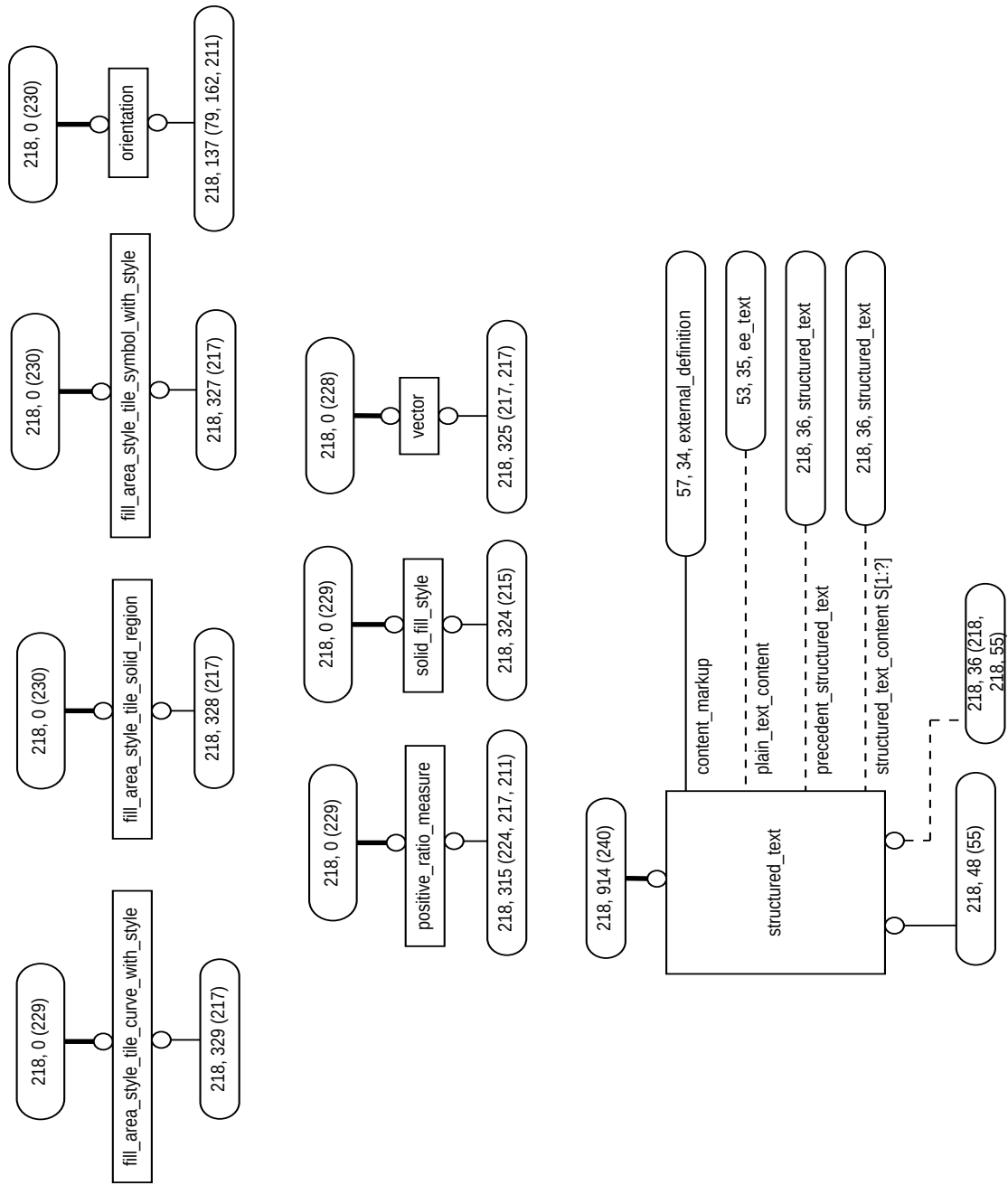


Figure H.218 – ARM EXPRESS-G Graphic 218 of 243

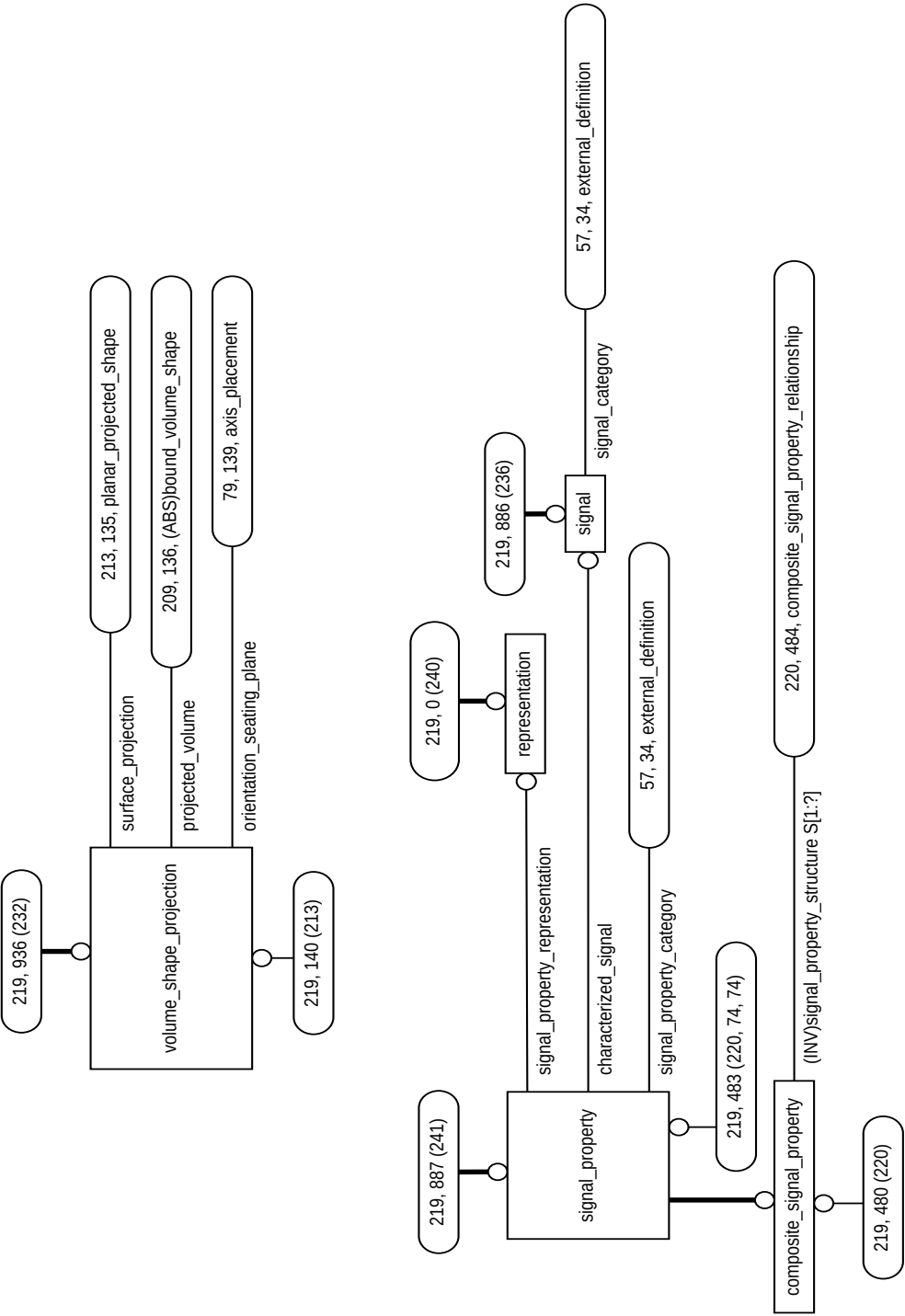


Figure H.219 – ARM EXPRESS-G Graphic 219 of 243

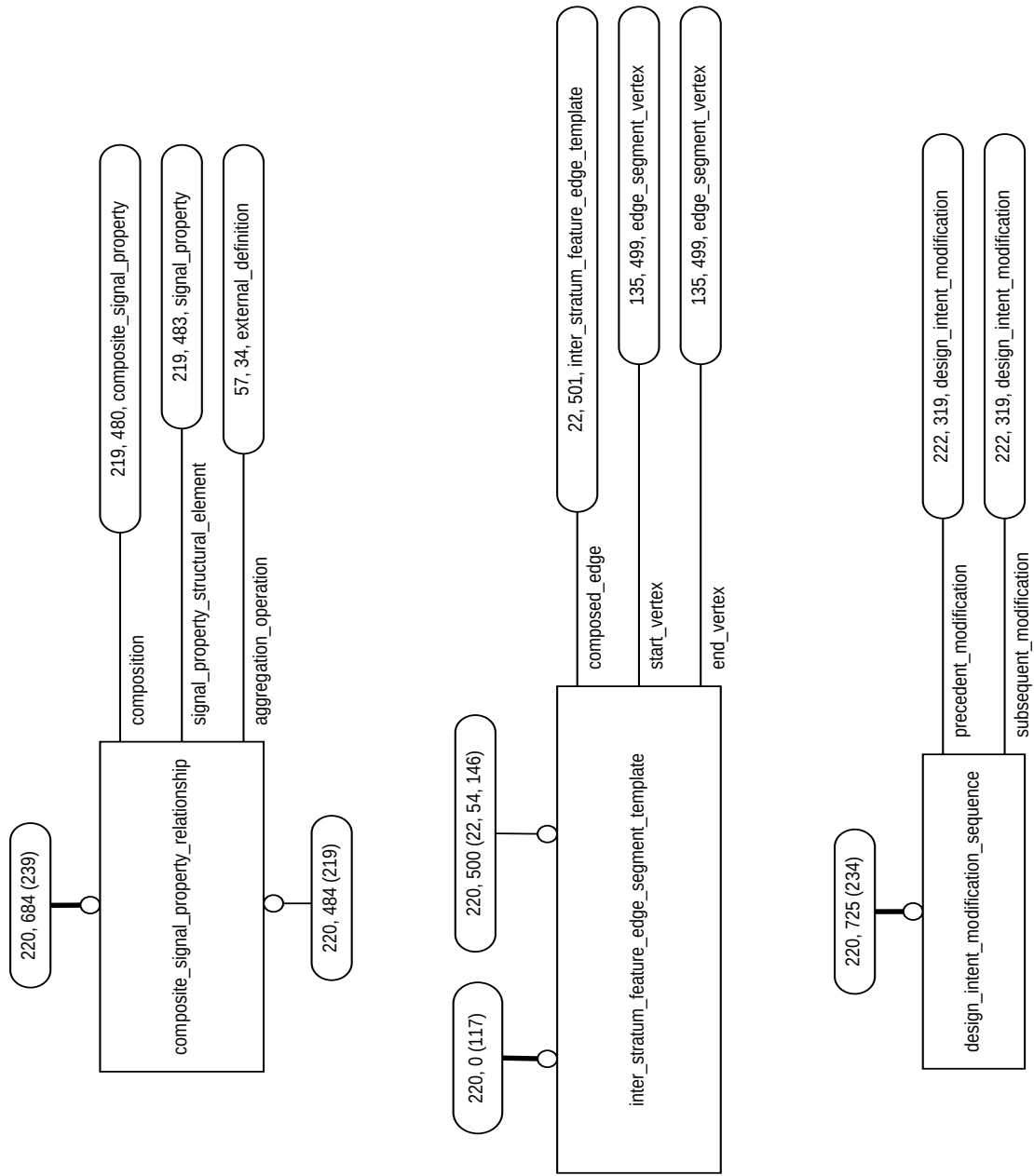


Figure H.220 – ARM EXPRESS-G Graphic 220 of 243

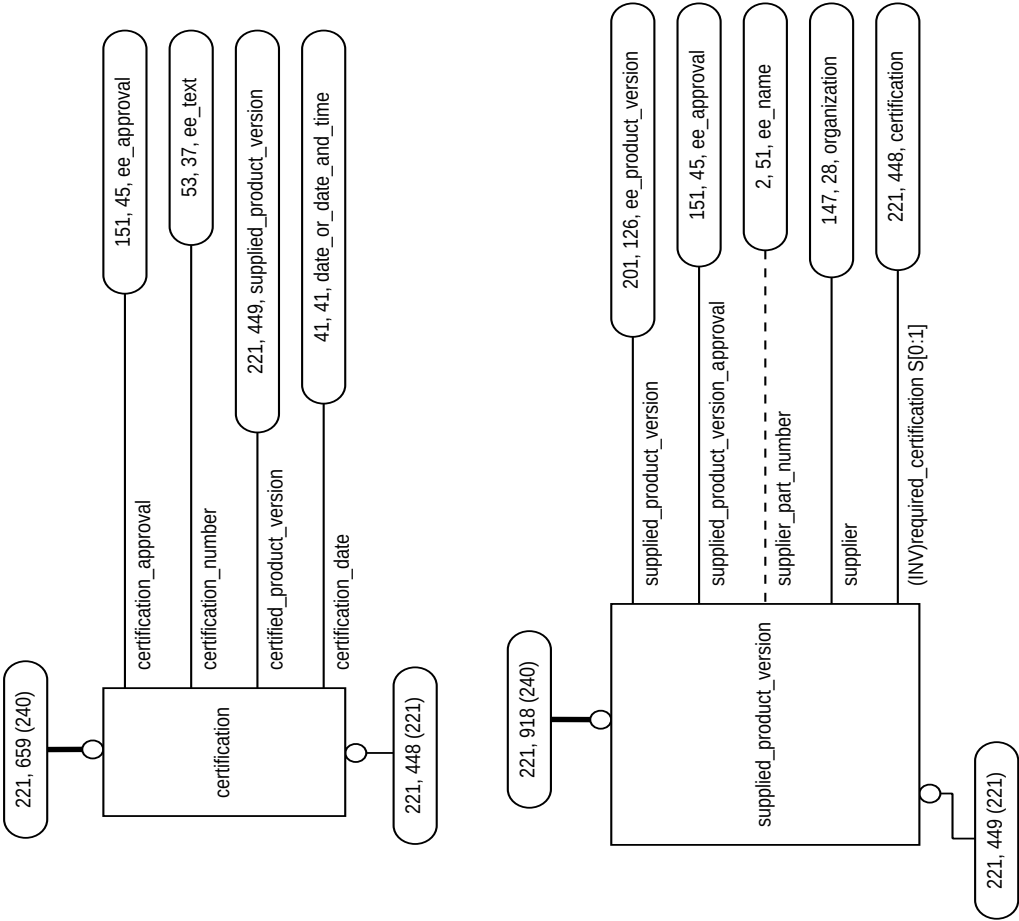


Figure H.221 – ARM EXPRESS-G Graphic 221 of 243

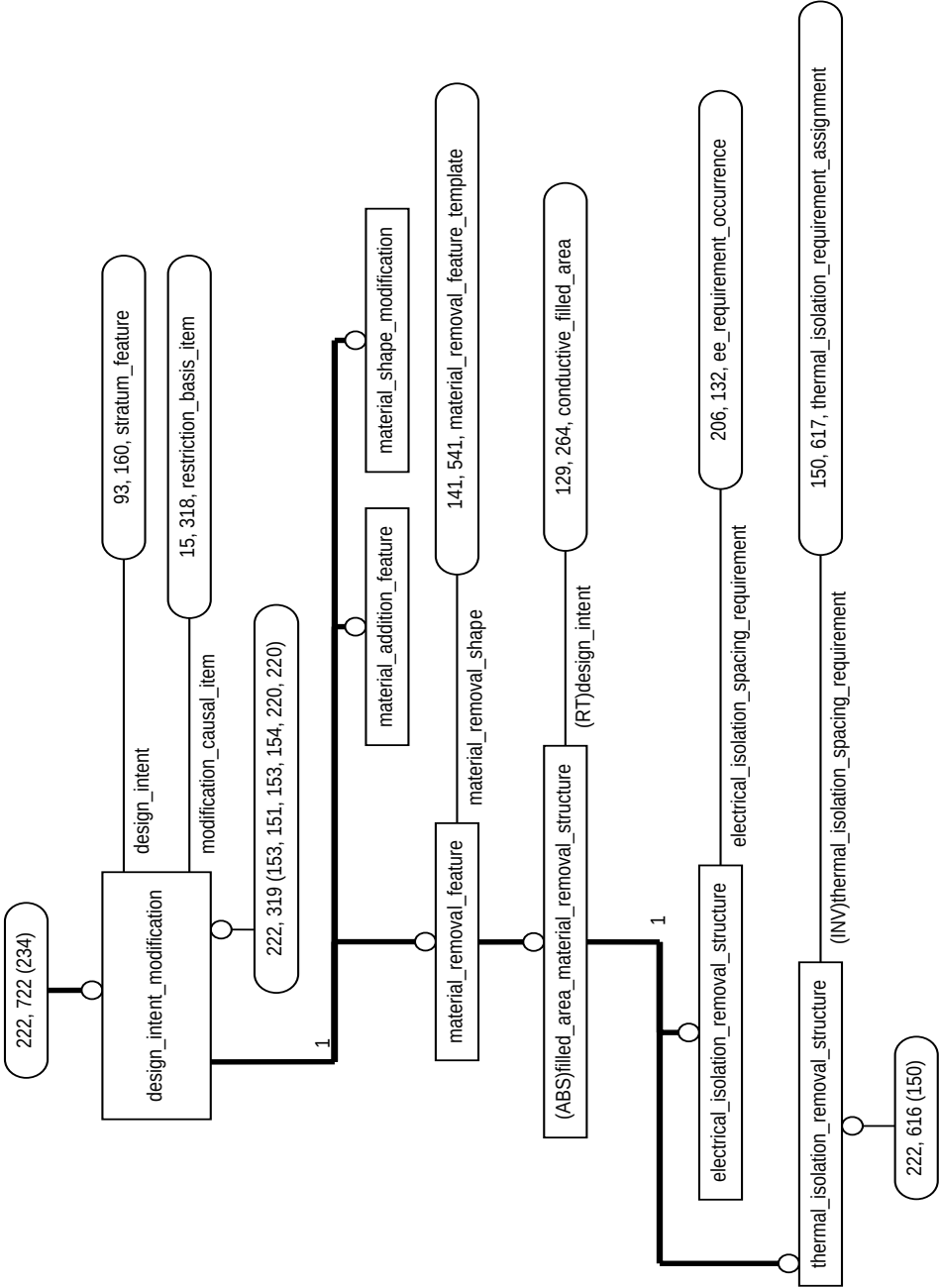


Figure H.222 – ARM EXPRESS-G Graphic 222 of 243

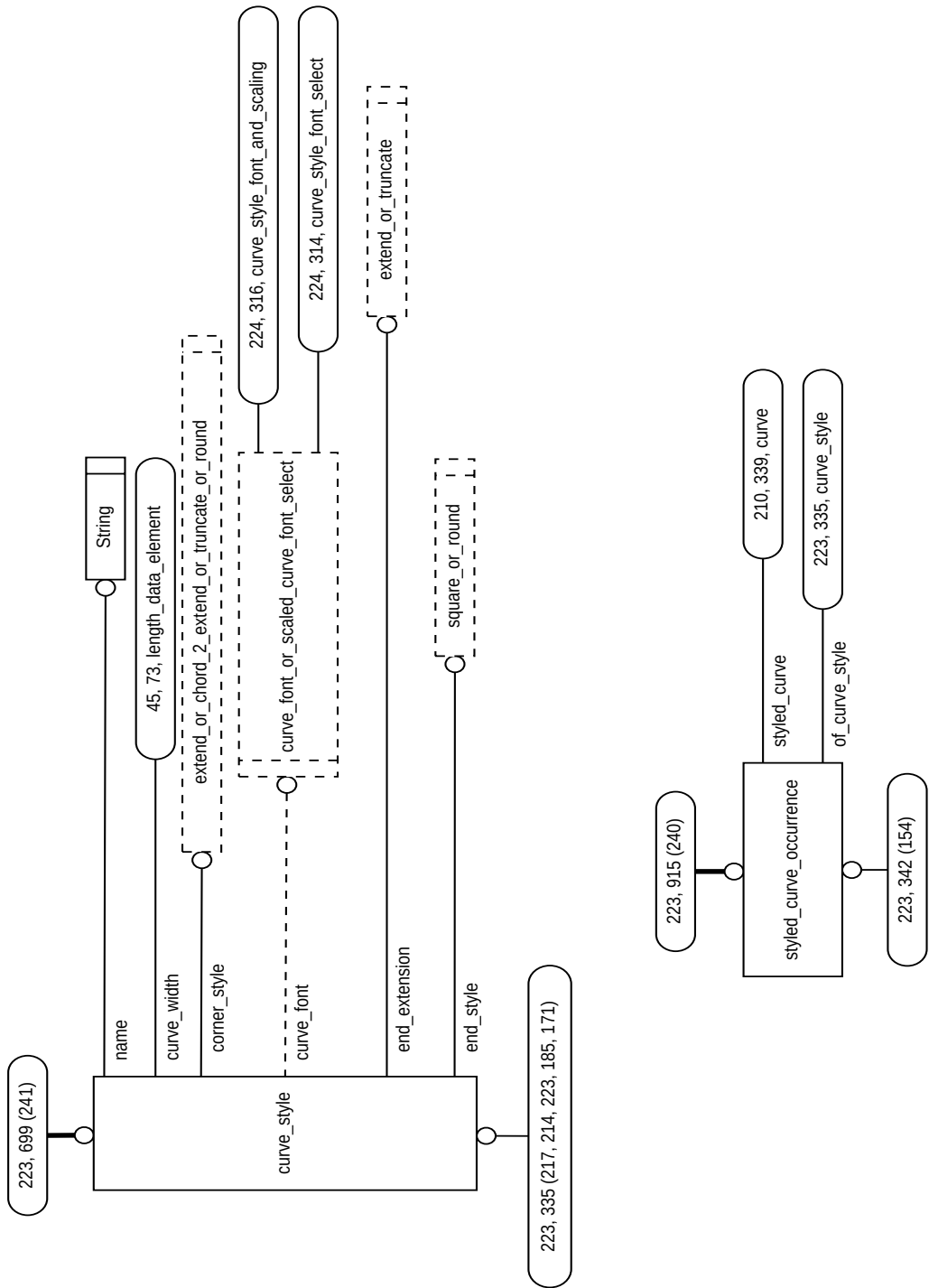


Figure H.223 – ARM EXPRESS-G Graphic 223 of 243

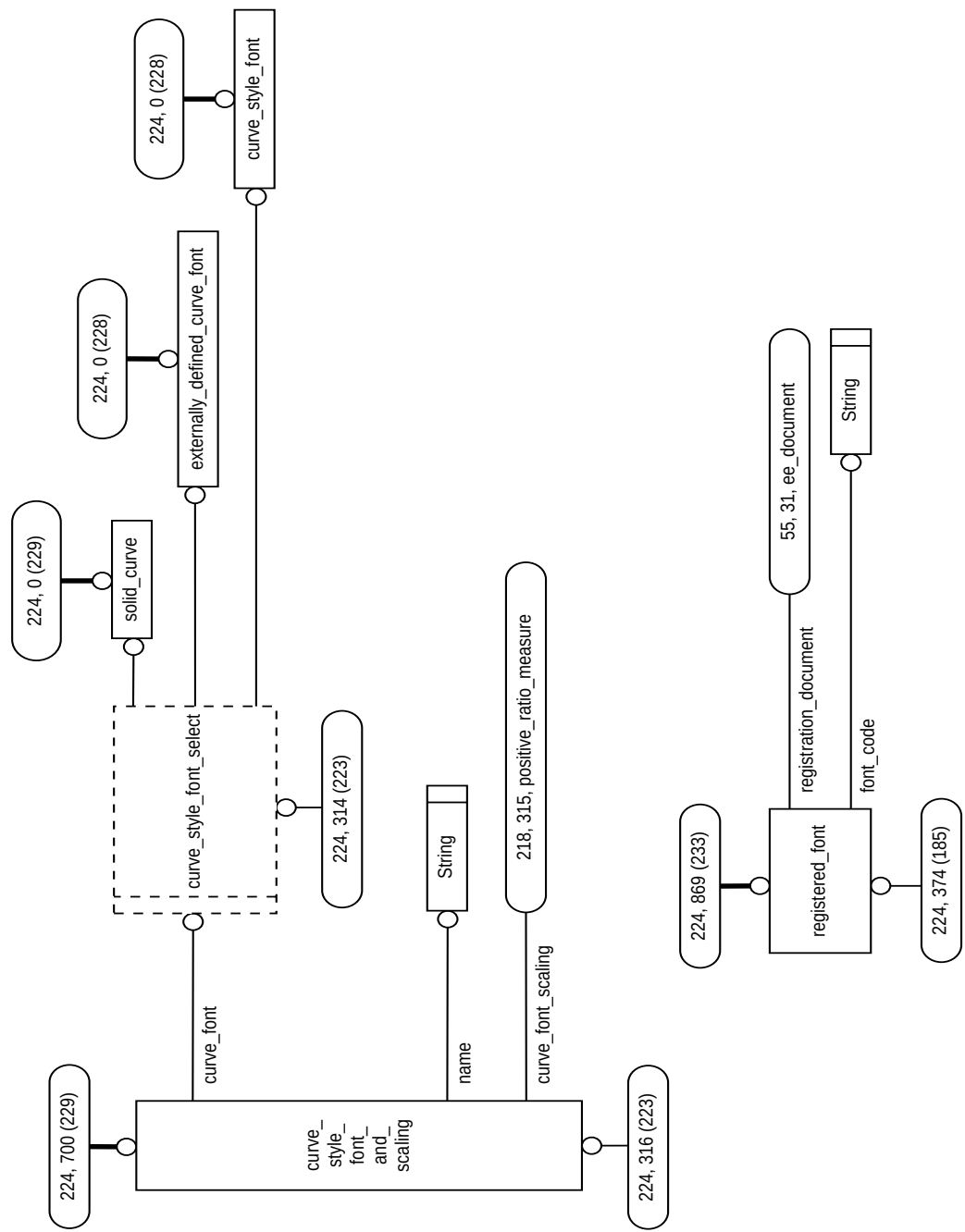


Figure H.224 – ARM EXPRESS-G Graphic 224 of 243

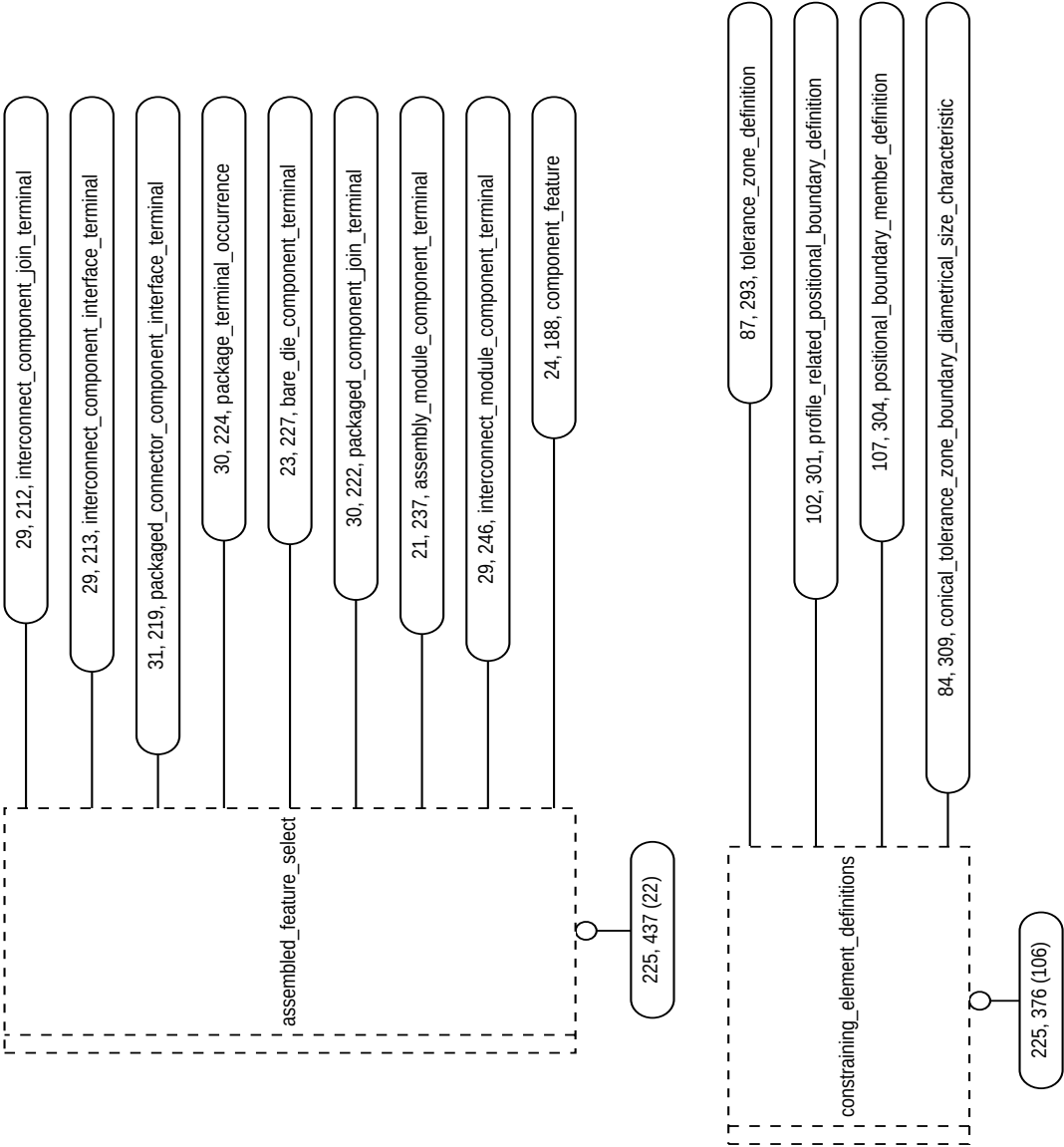


Figure H.225 – ARM EXPRESS-G Graphic 225 of 243

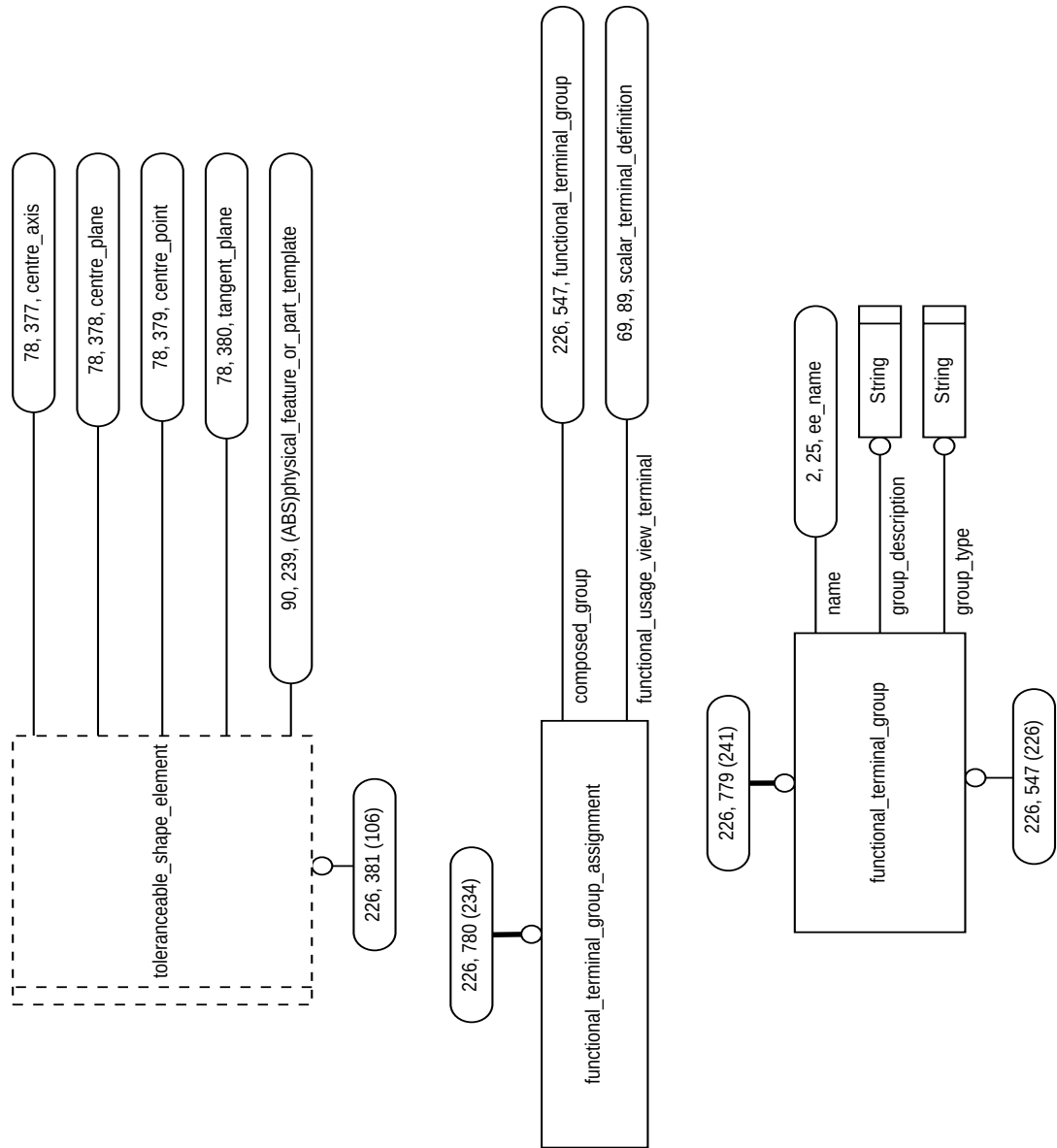


Figure H.226 – ARM EXPRESS-G Graphic 226 of 243

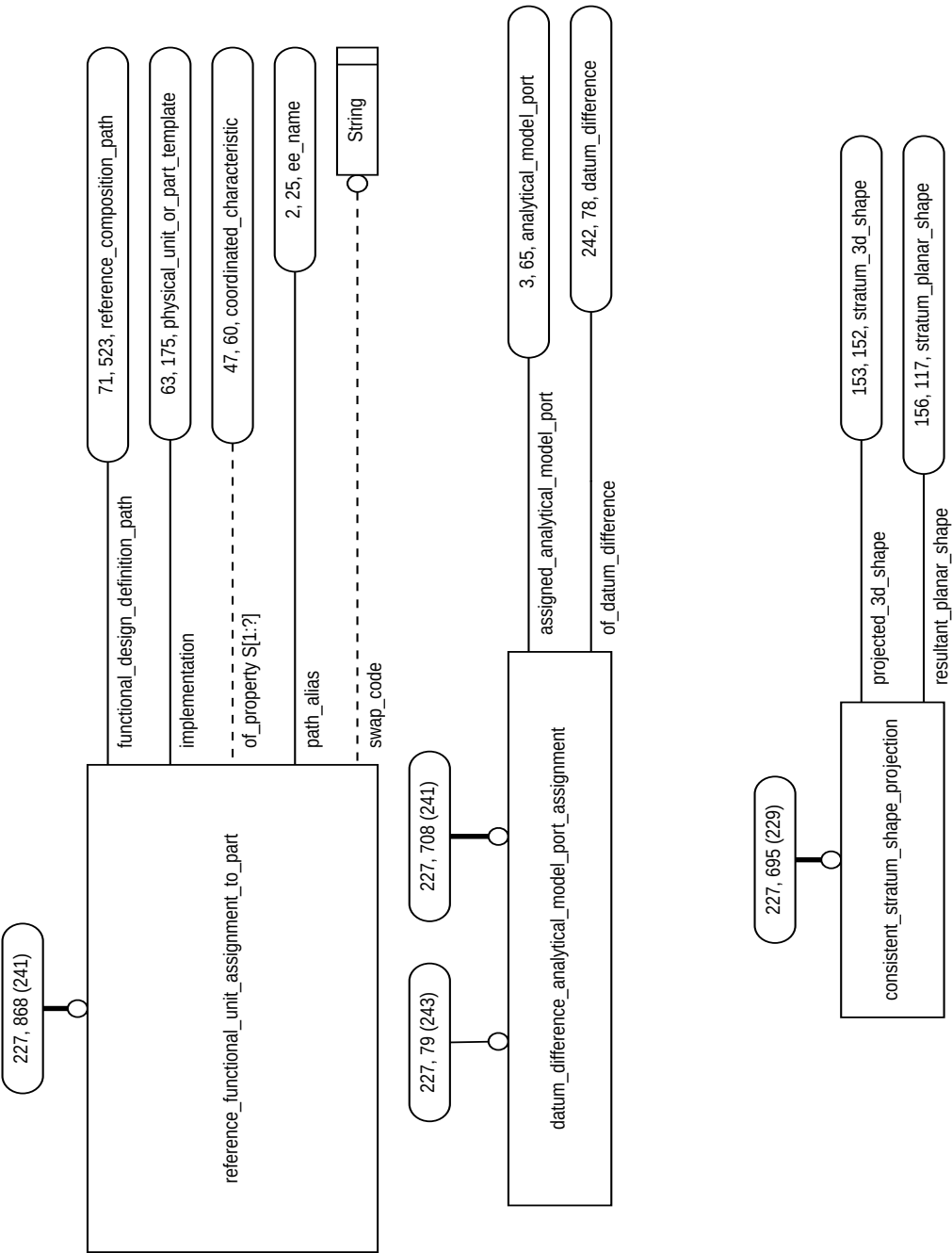


Figure H.227 – ARM EXPRESS-G Graphic 227 of 243

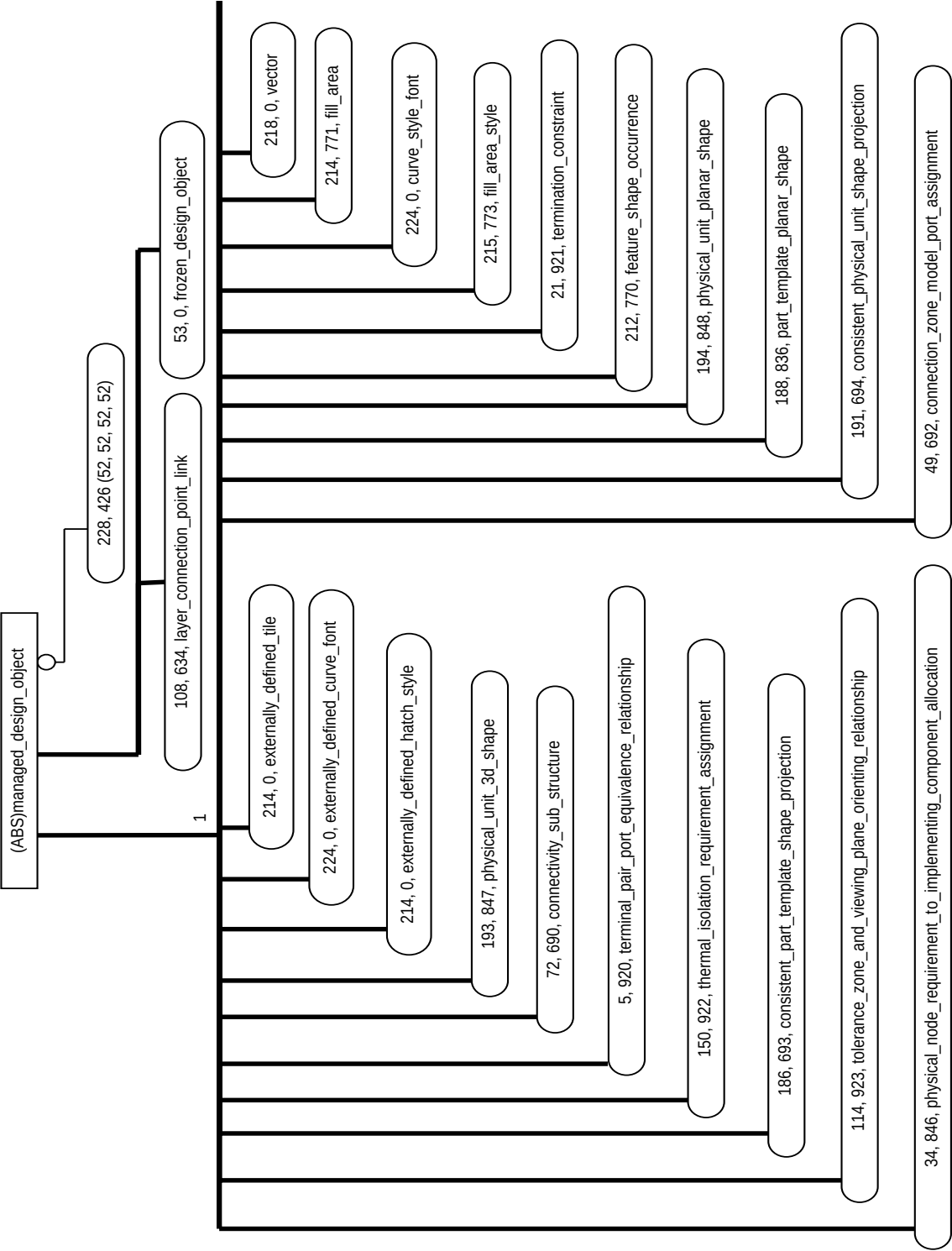


Figure H.228 – ARM EXPRESS-G Graphic 228 of 243

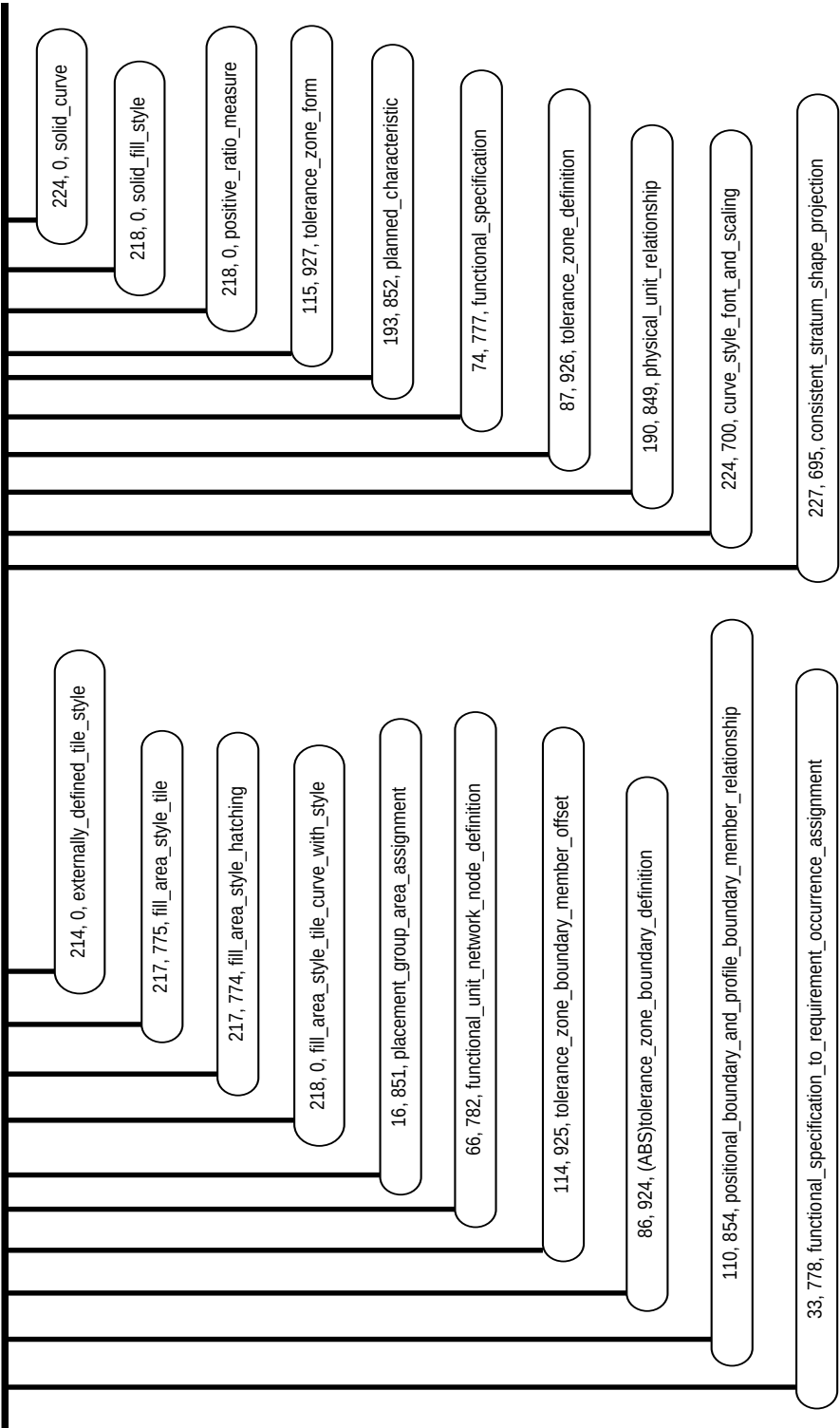


Figure H.229 – ARM EXPRESS-G Graphic 229 of 243

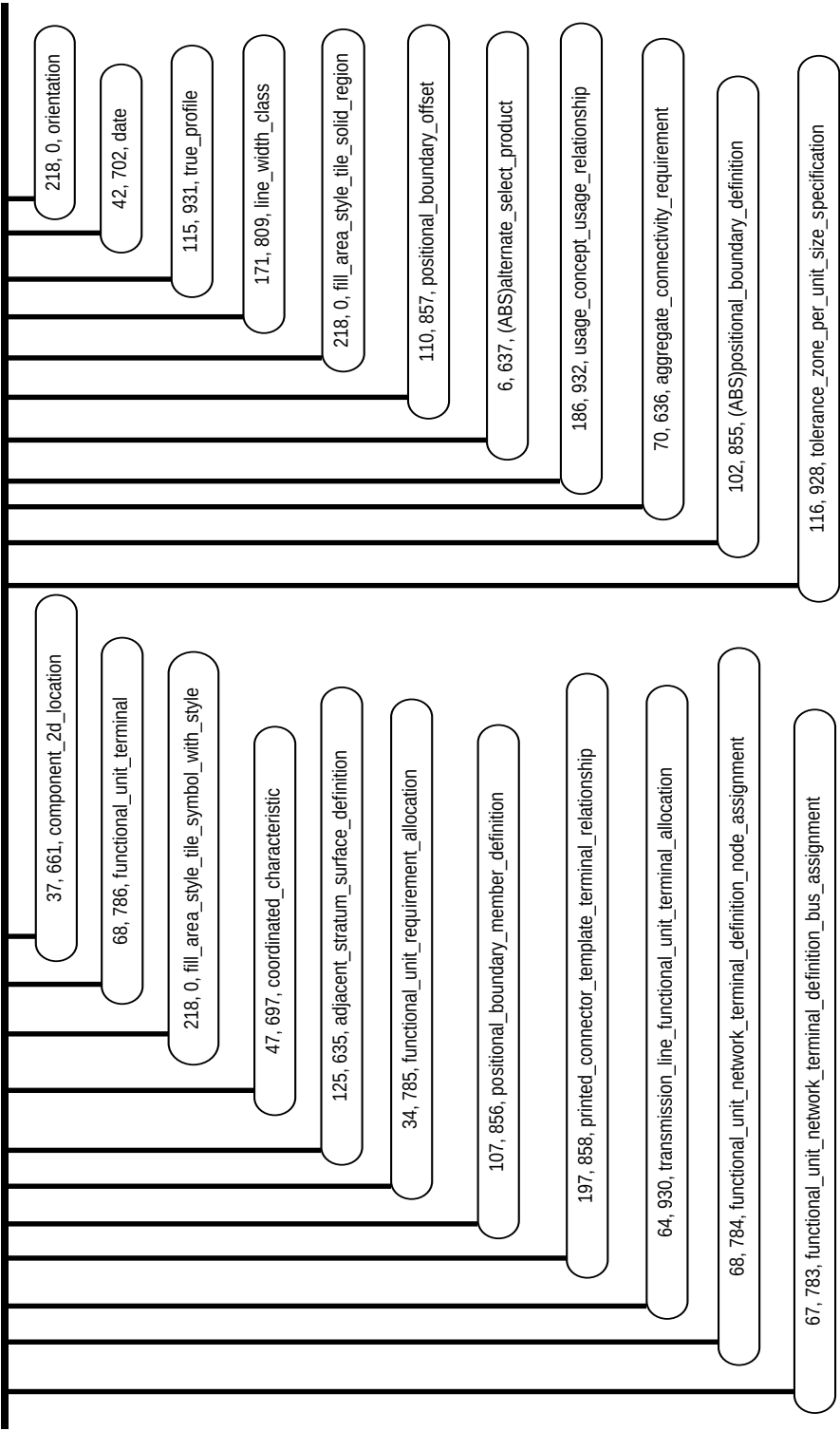


Figure H.230 – ARM EXPRESS-G Graphic 230 of 243

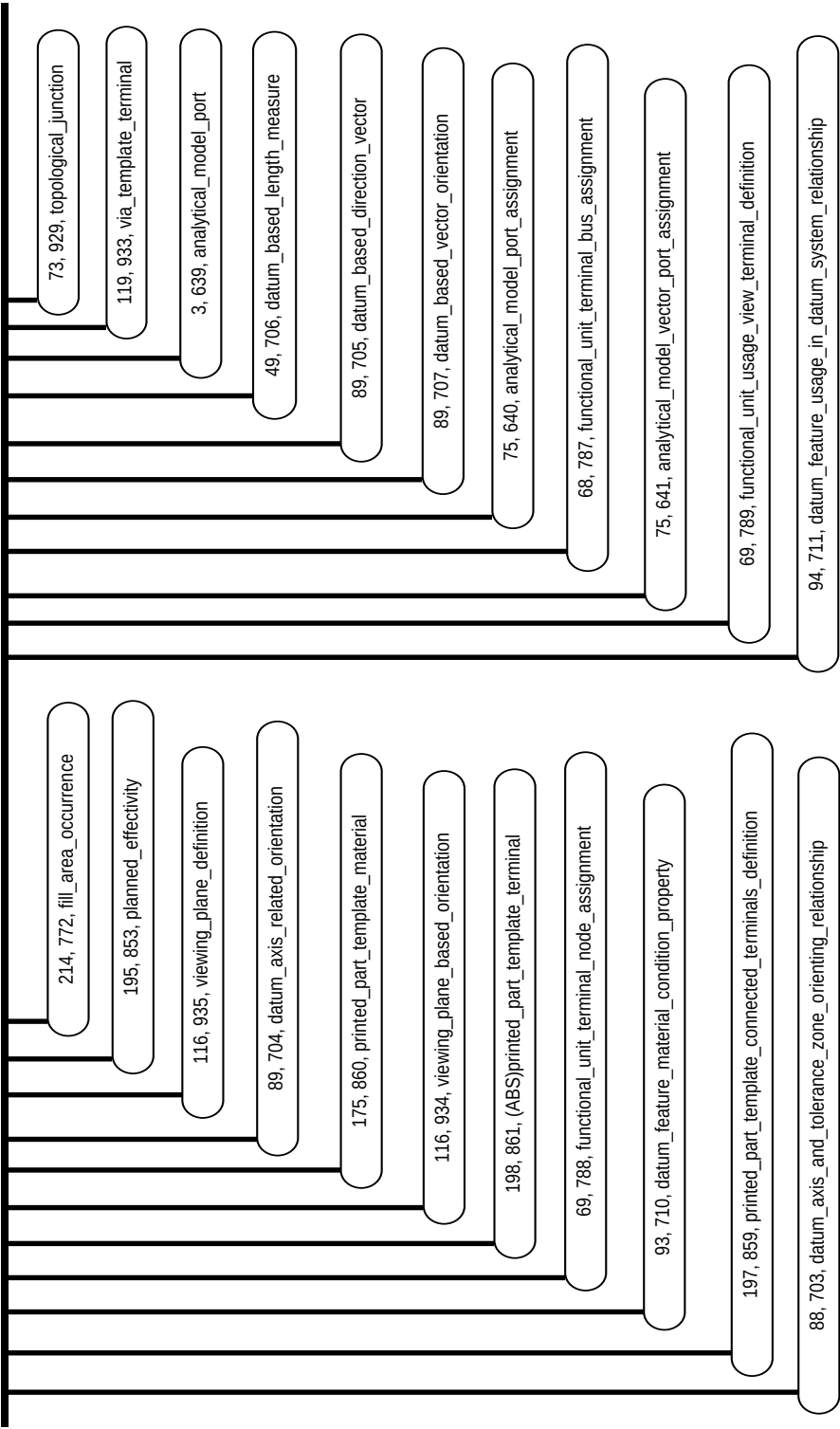


Figure H.231 – ARM EXPRESS-G Graphic 231 of 243

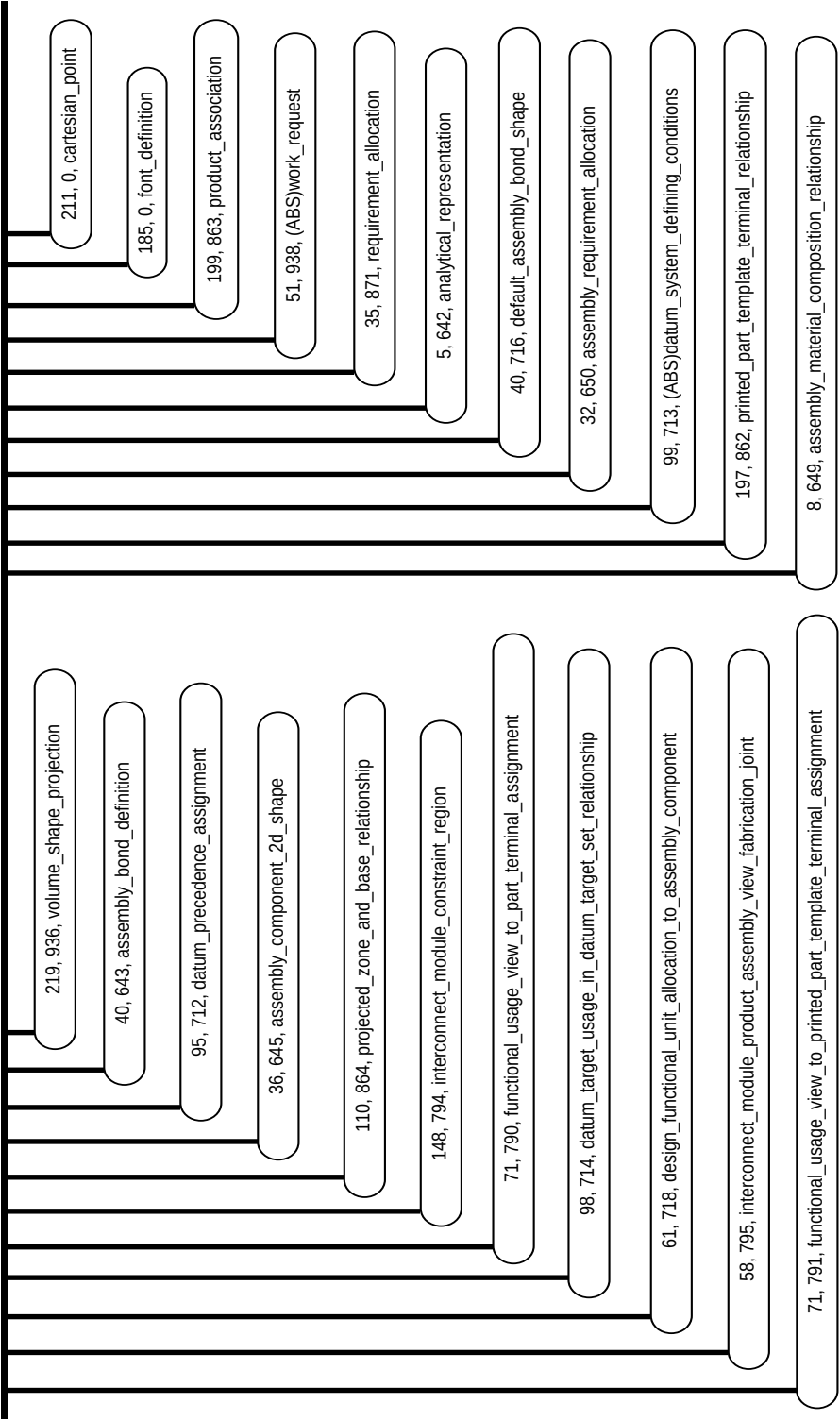


Figure H.232 – ARM EXPRESS-G Graphic 232 of 243

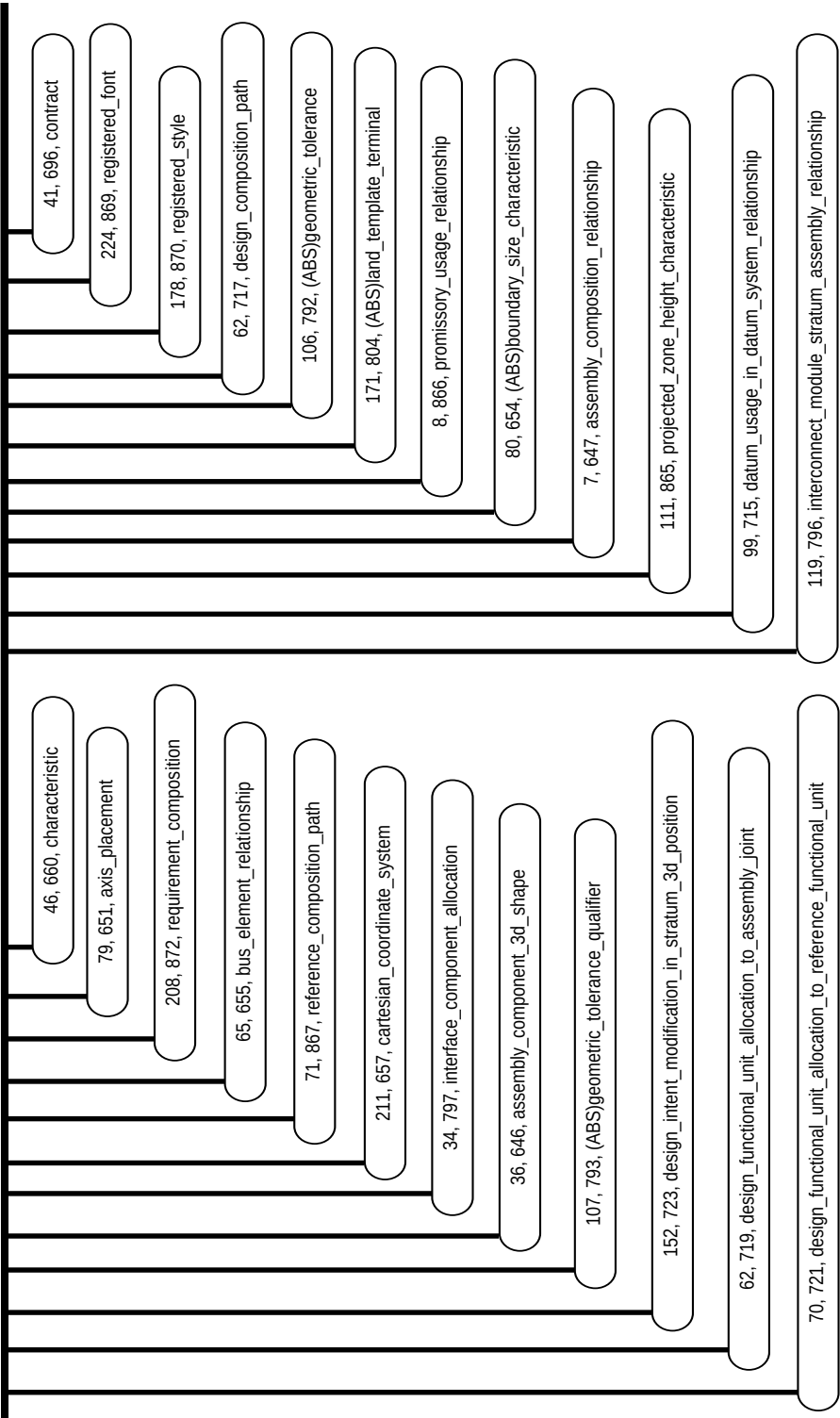


Figure H.233 – ARM EXPRESS-G Graphic 233 of 243

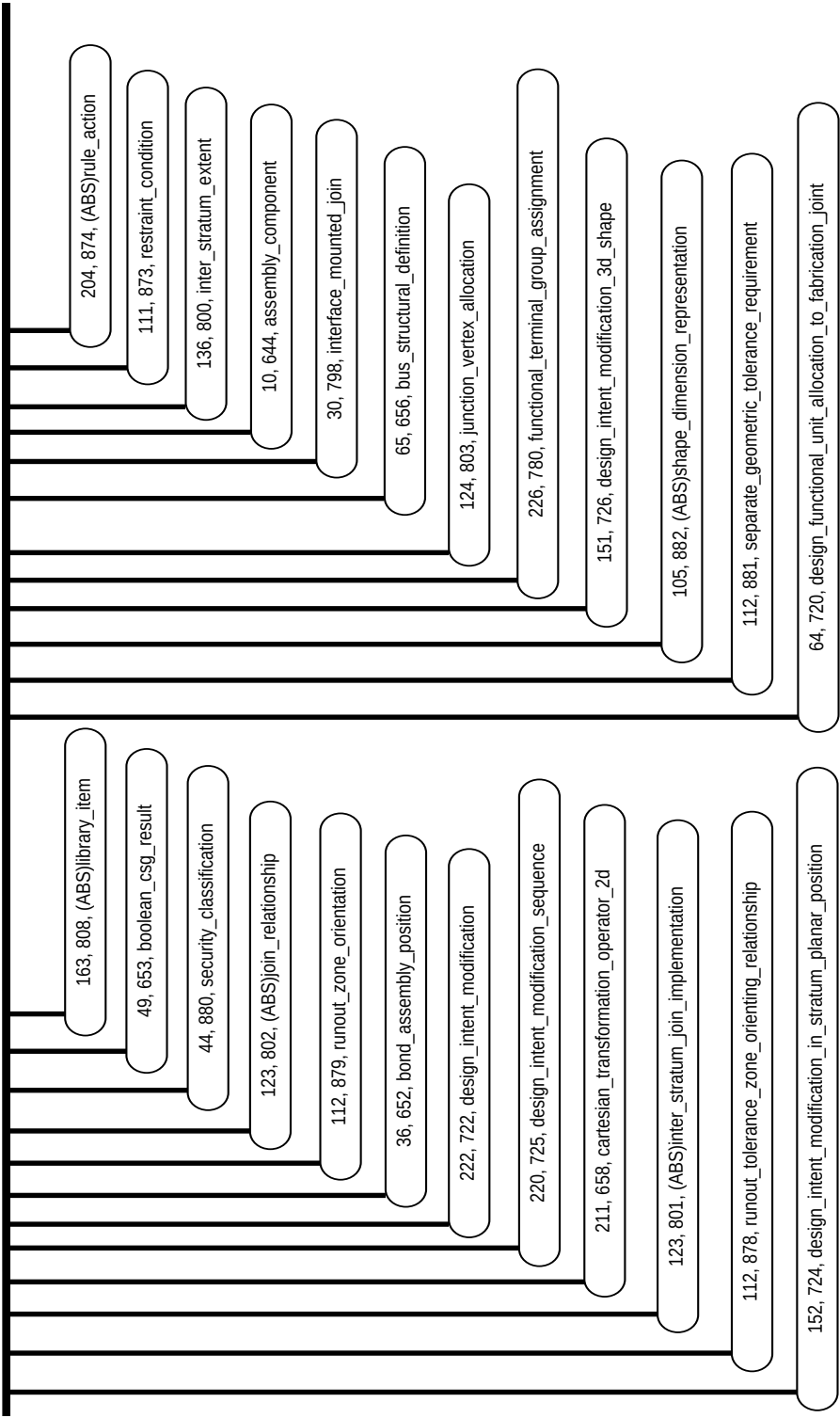


Figure H.234 – ARM EXPRESS-G Graphic 234 of 243

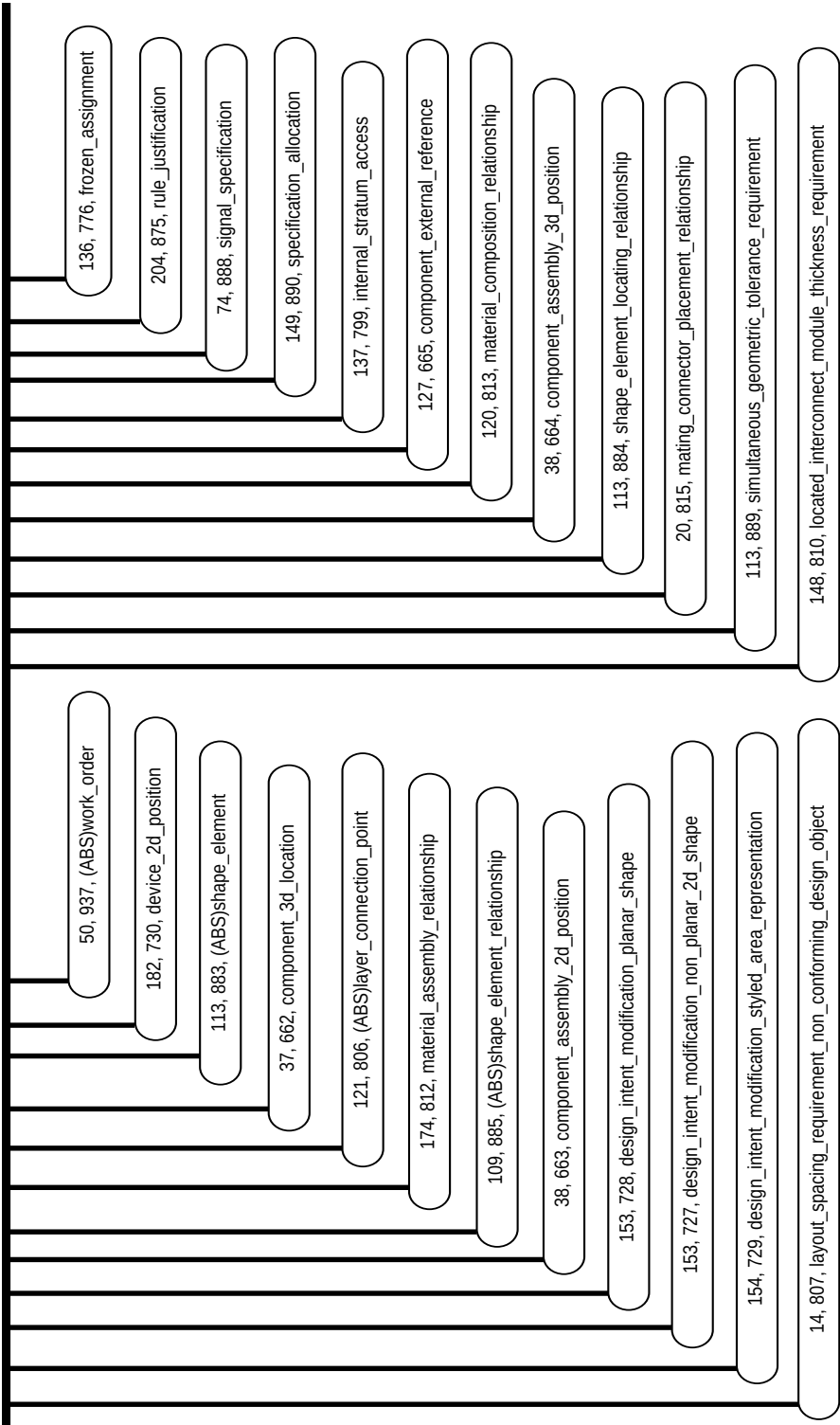


Figure H.235 – ARM EXPRESS-G Graphic 235 of 243



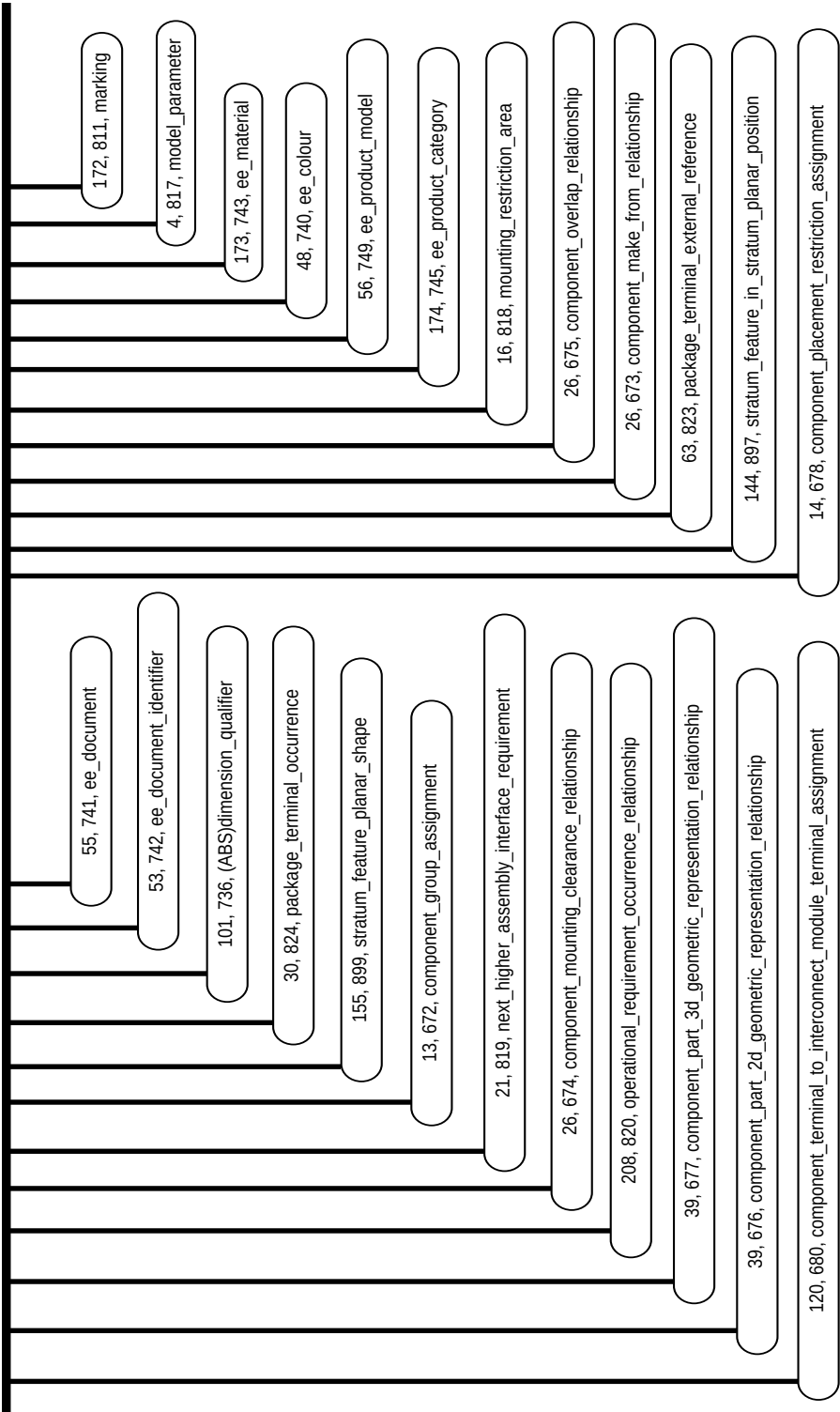


Figure H.237 – ARM EXPRESS-G Graphic 237 of 243

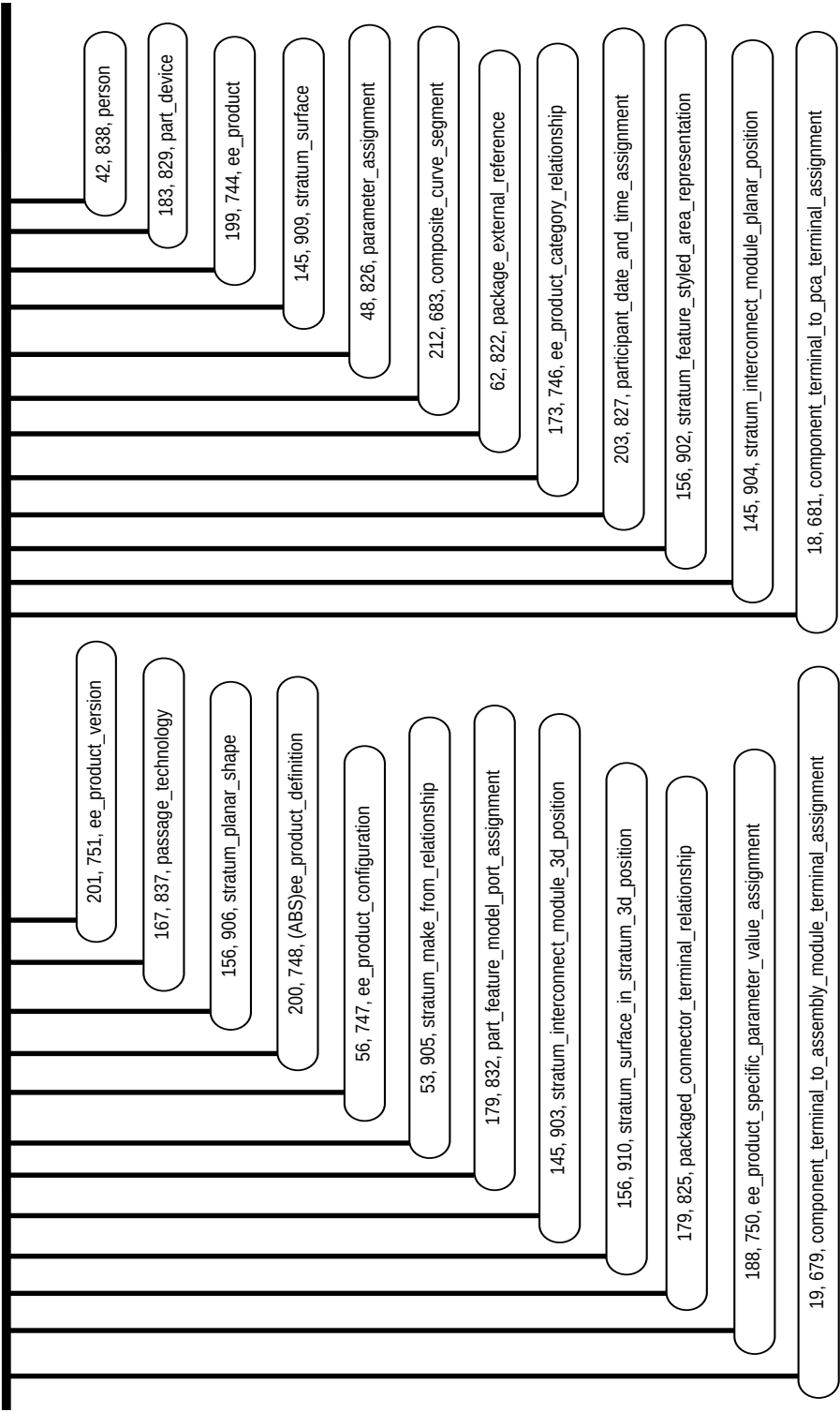


Figure H.238 – ARM EXPRESS-G Graphic 238 of 243

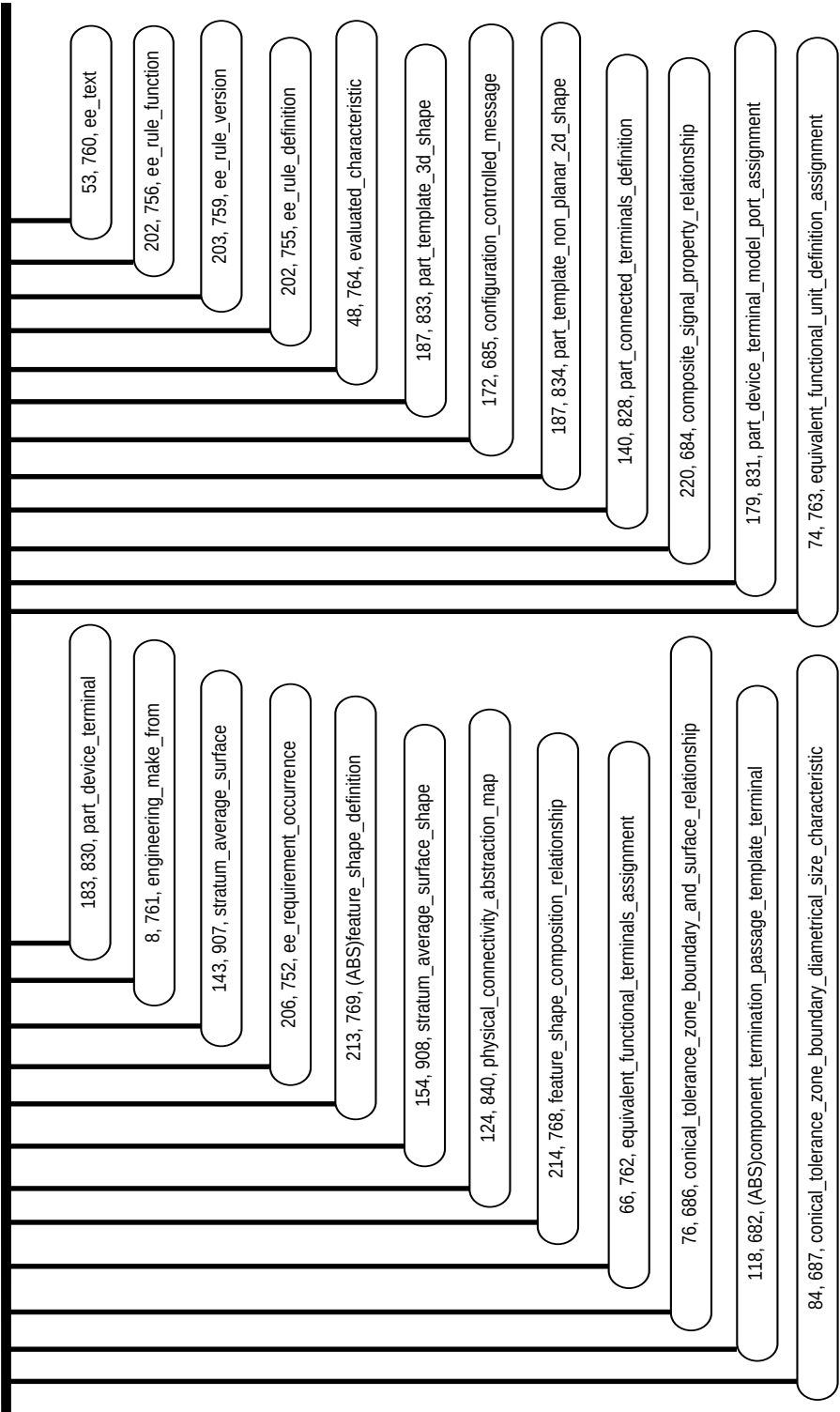


Figure H.239 – ARM EXPRESS-G Graphic 239 of 243

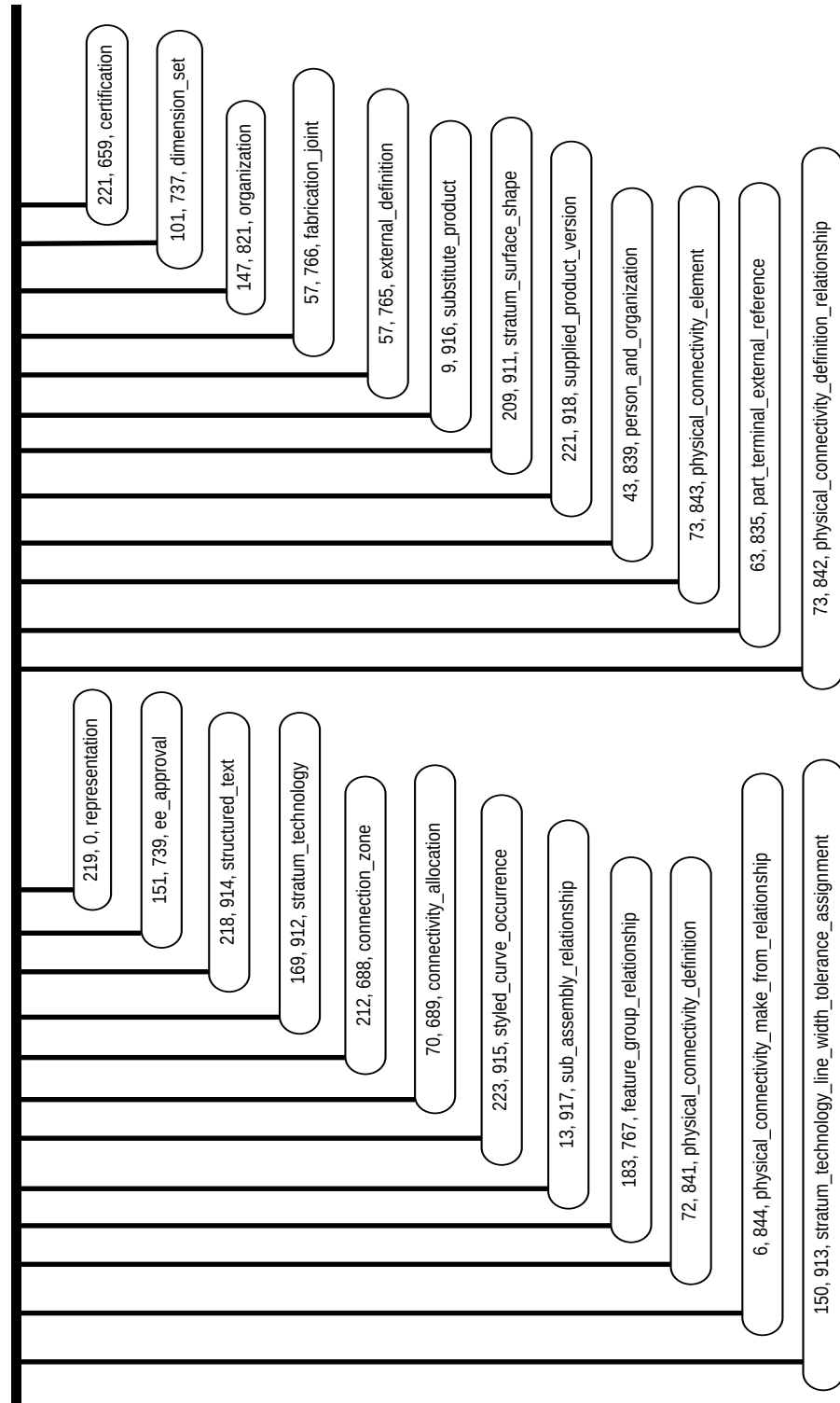


Figure H.240 – ARM EXPRESS-G Graphic 240 of 243

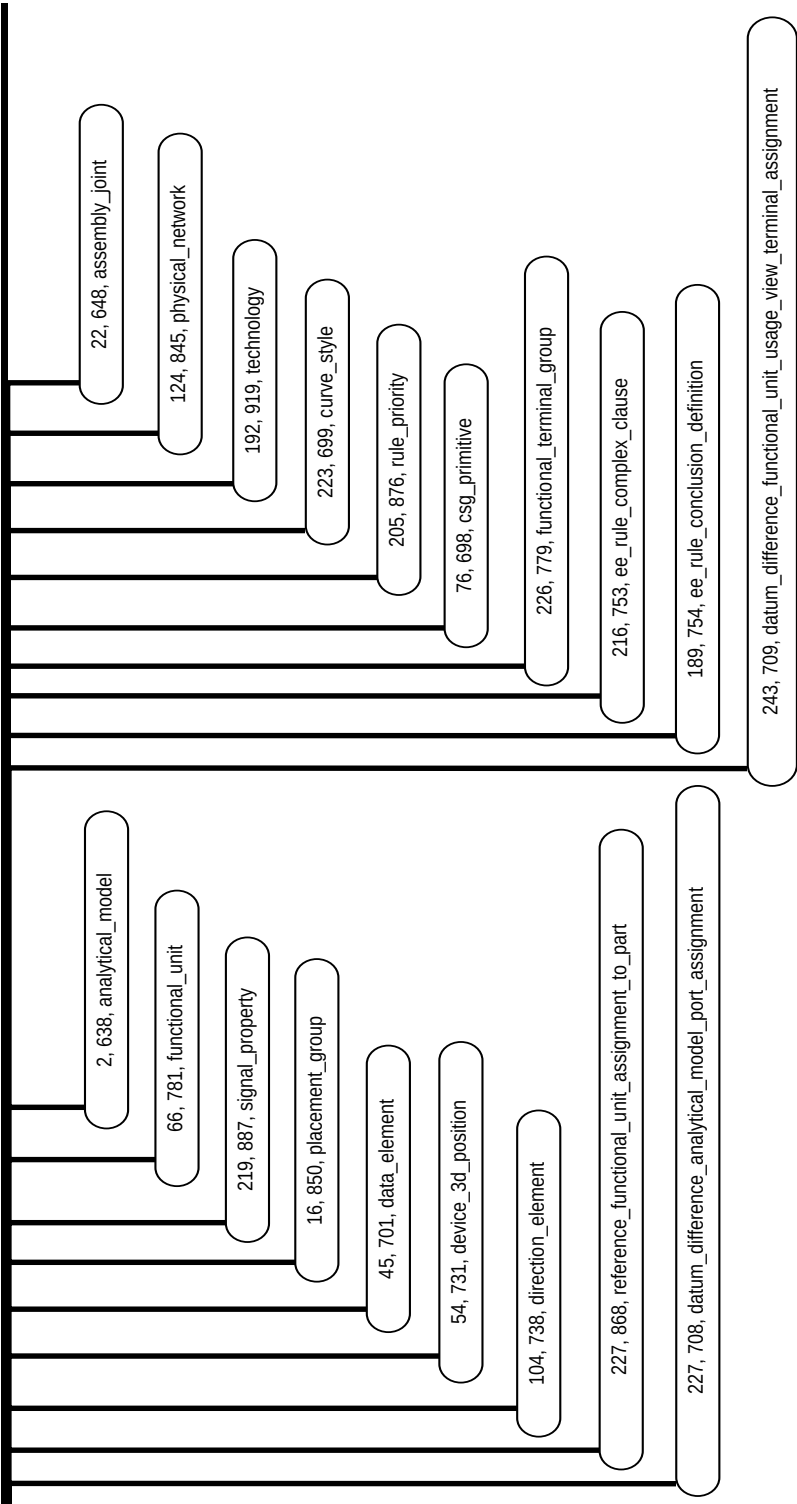


Figure H.241 – ARM EXPRESS-G Graphic 241 of 243

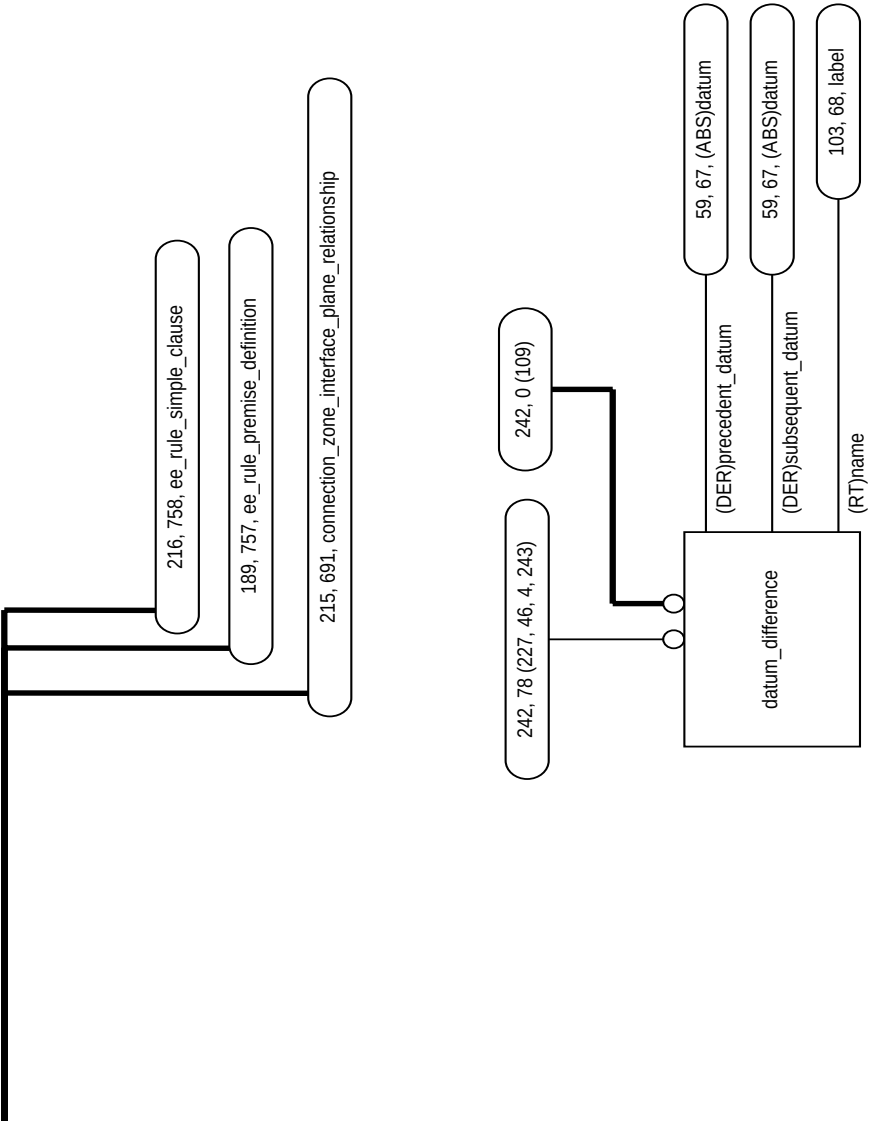


Figure H.242 – ARM EXPRESS-G Graphic 242 of 243

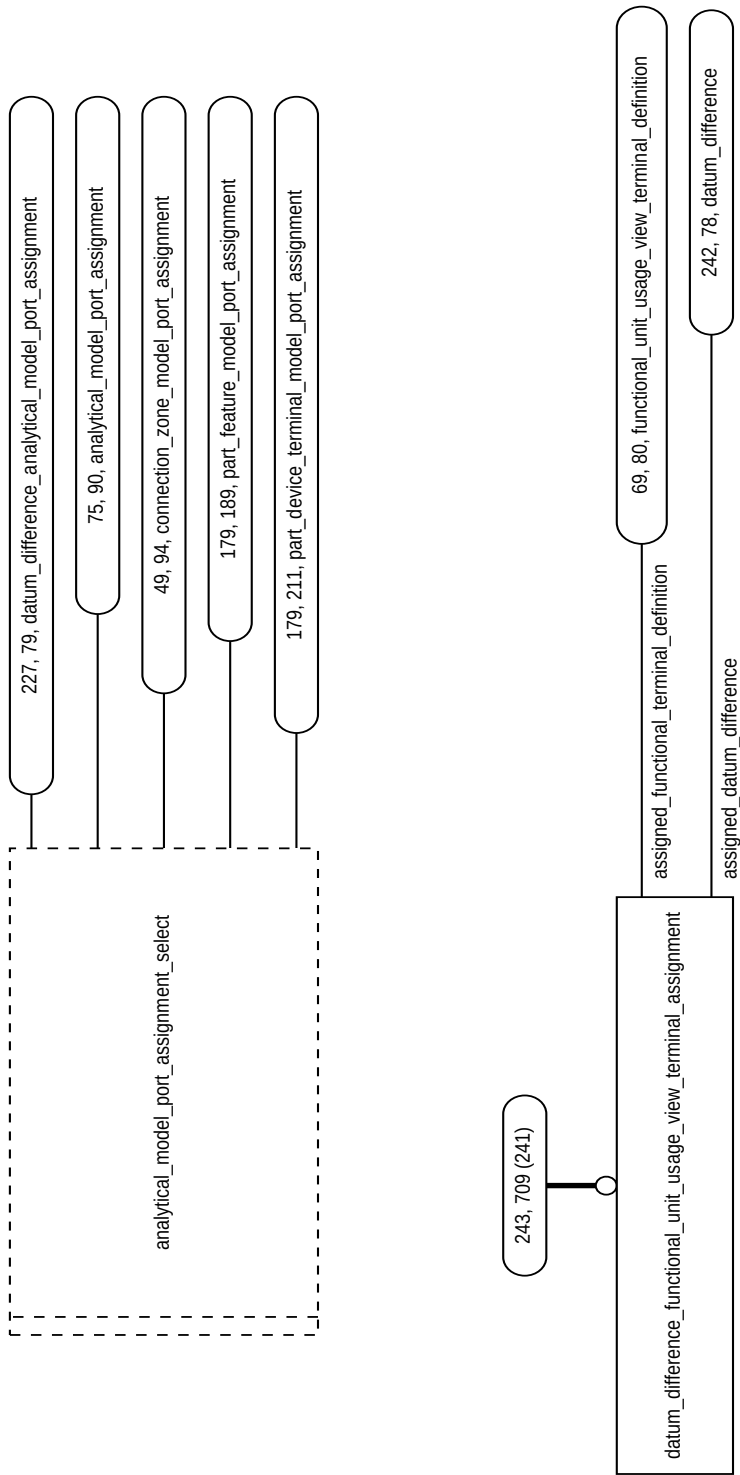


Figure H.243 – ARM EXPRESS-G Graphic 243 of 243

Annex J
(informative)
AIM EXPRESS-G

Figures J.1 through J.80 correspond to the AIM EXPRESS expanded listing given in annex A. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

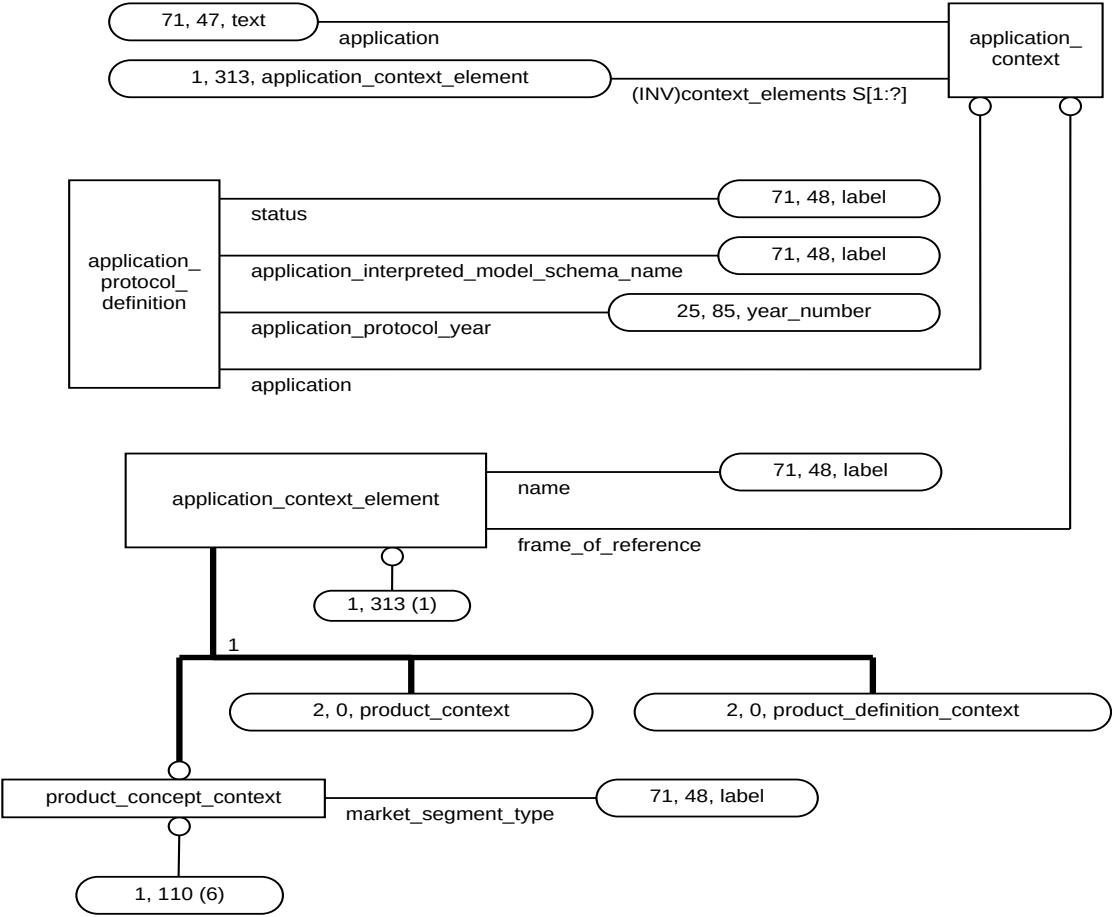


Figure J.1 – AIM EXPRESS-G diagram 1 of 80

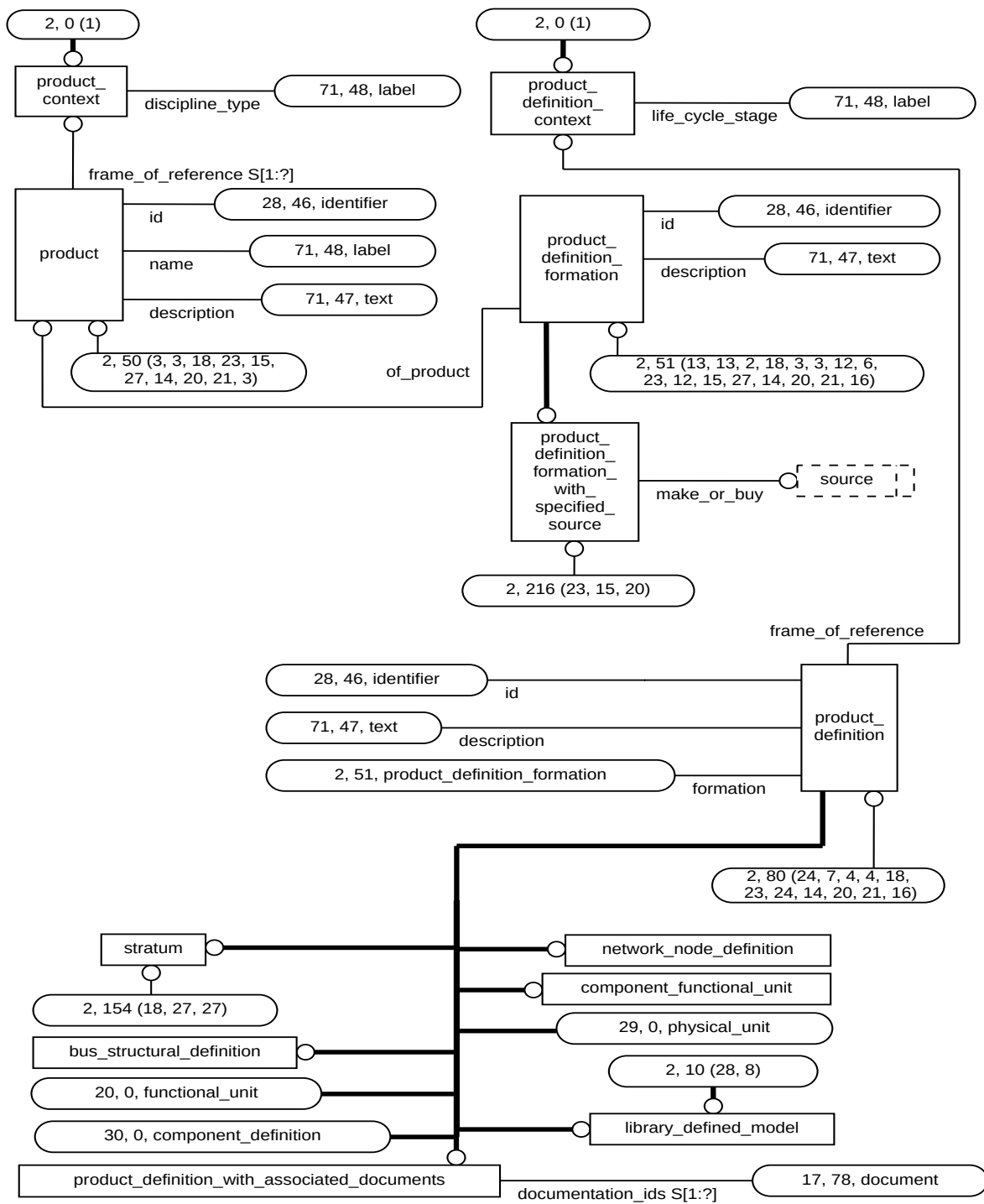


Figure J.2 – AIM EXPRESS-G diagram 2 of 80

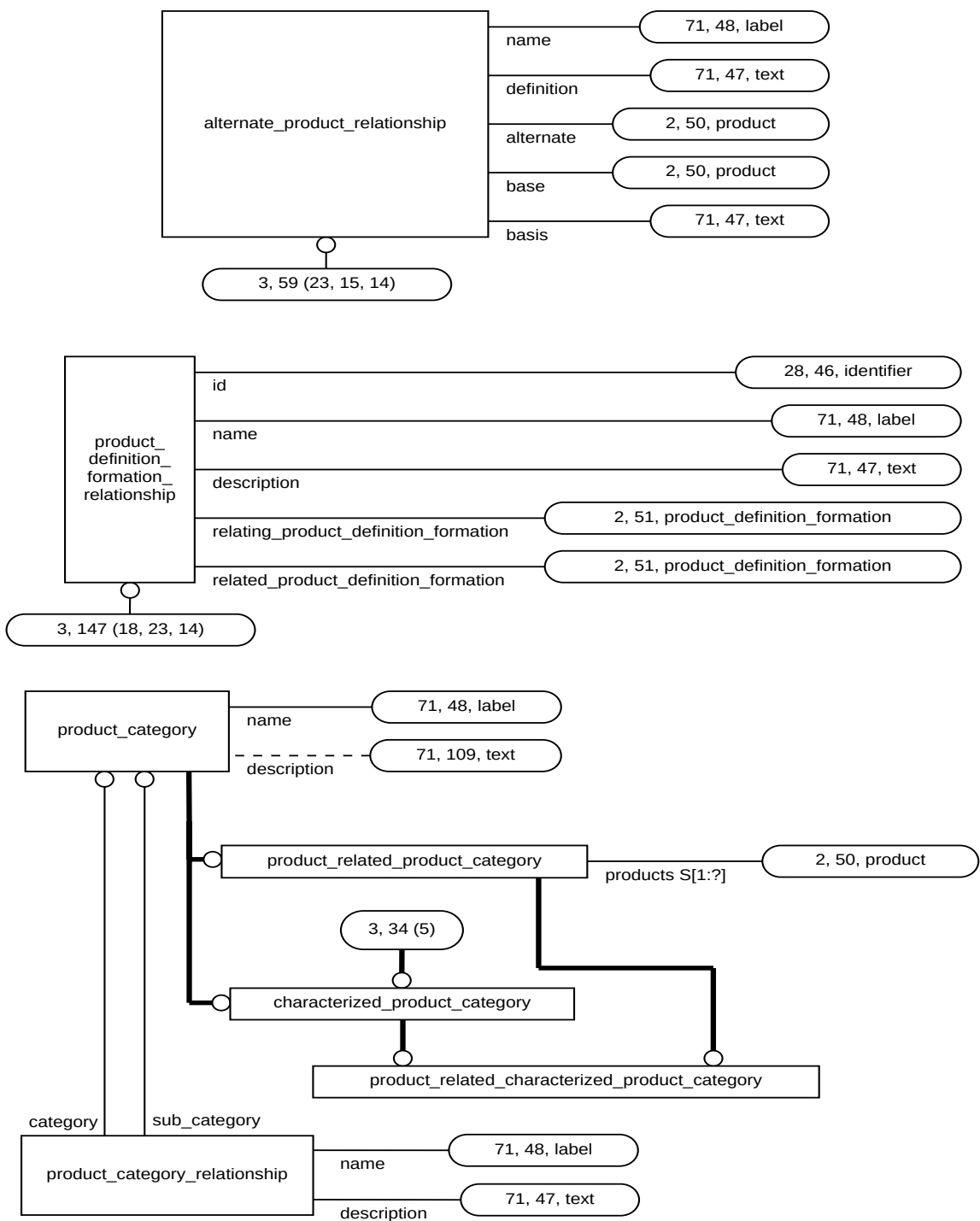


Figure J.3 – AIM EXPRESS-G diagram 3 of 80

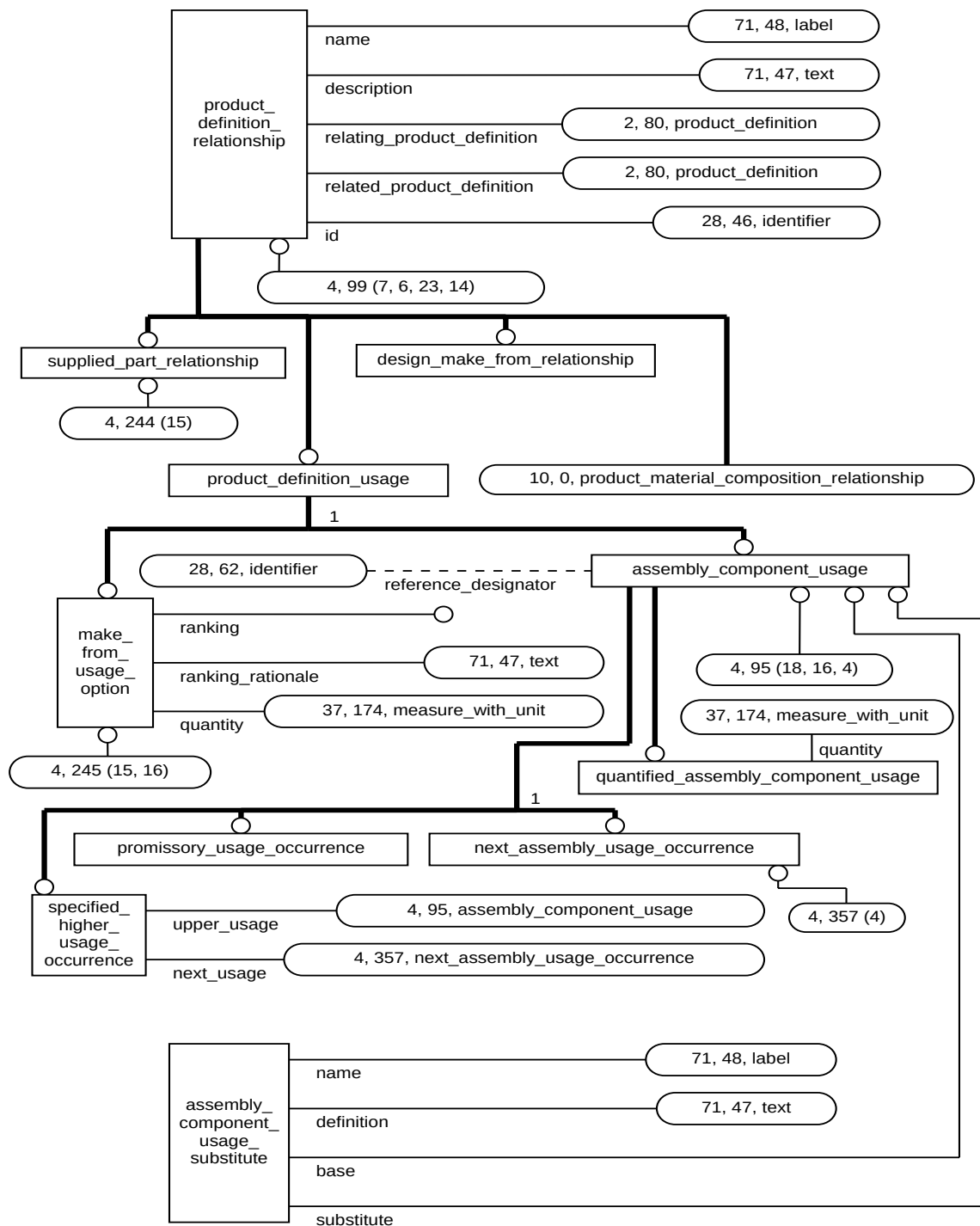


Figure J.4 – AIM EXPRESS-G diagram 4 of 80

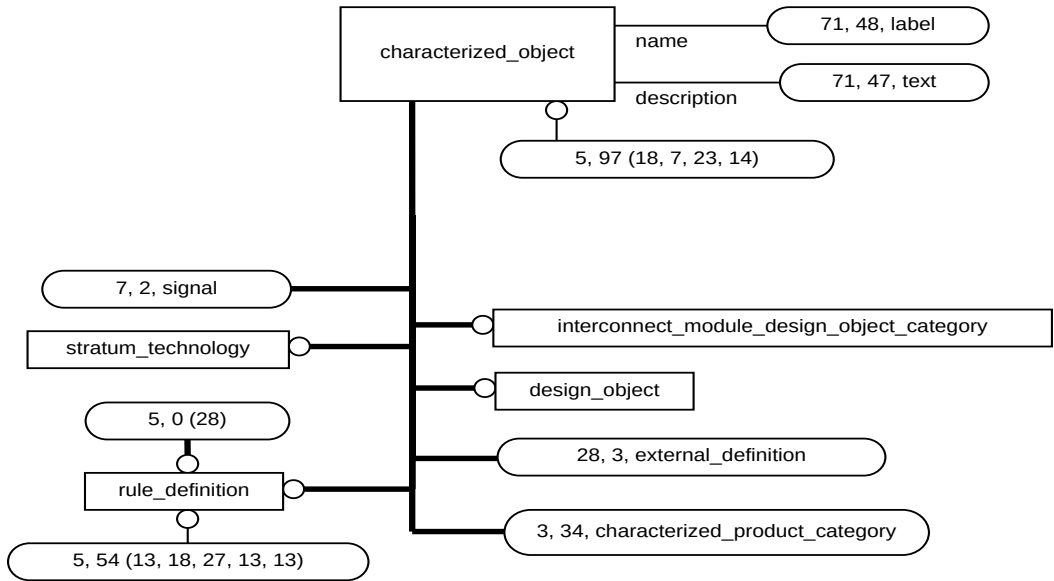


Figure J.5 – AIM EXPRESS-G diagram 5 of 80

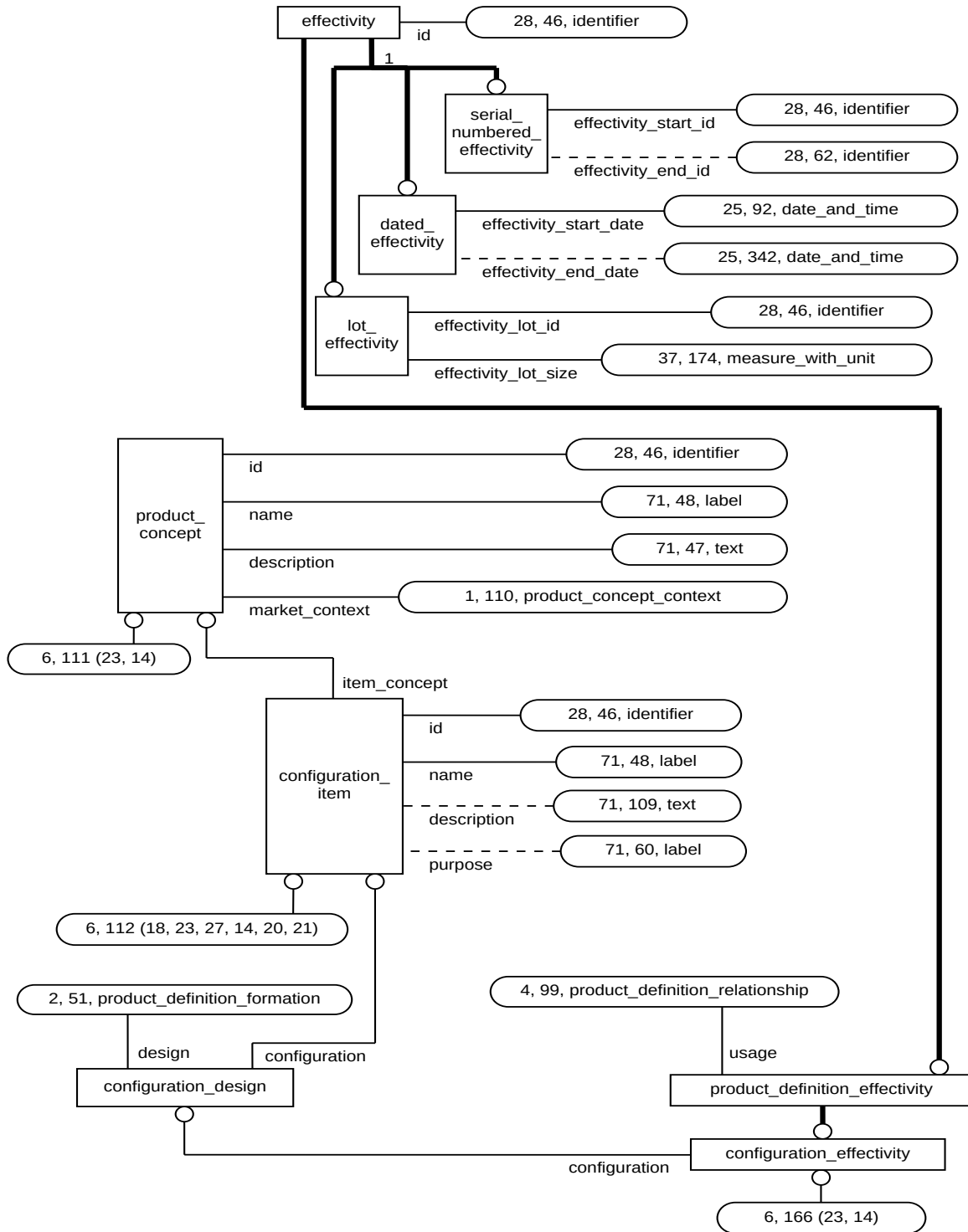


Figure J.6 – AIM EXPRESS-G diagram 6 of 80

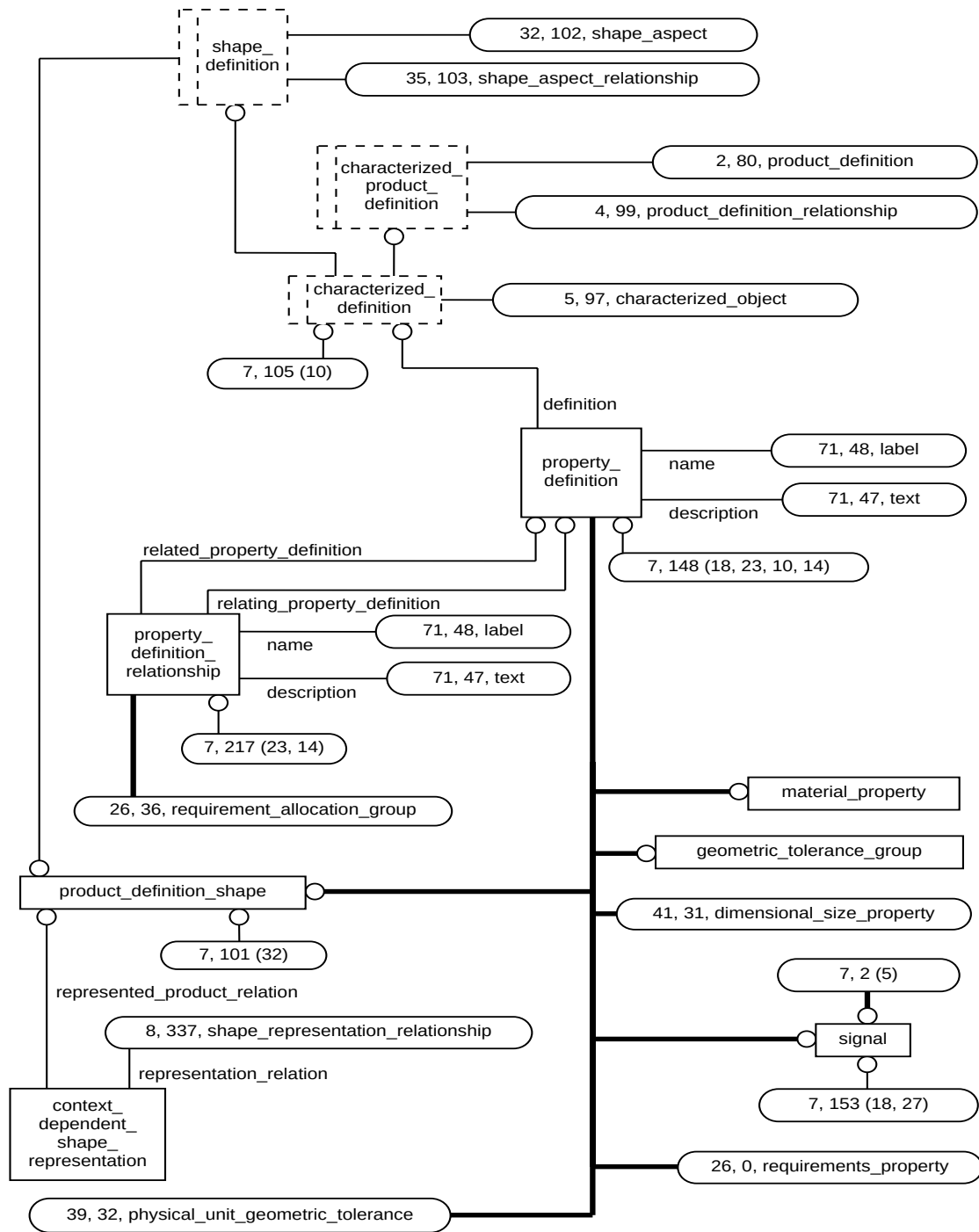


Figure J.7 – AIM EXPRESS-G diagram 7 of 80

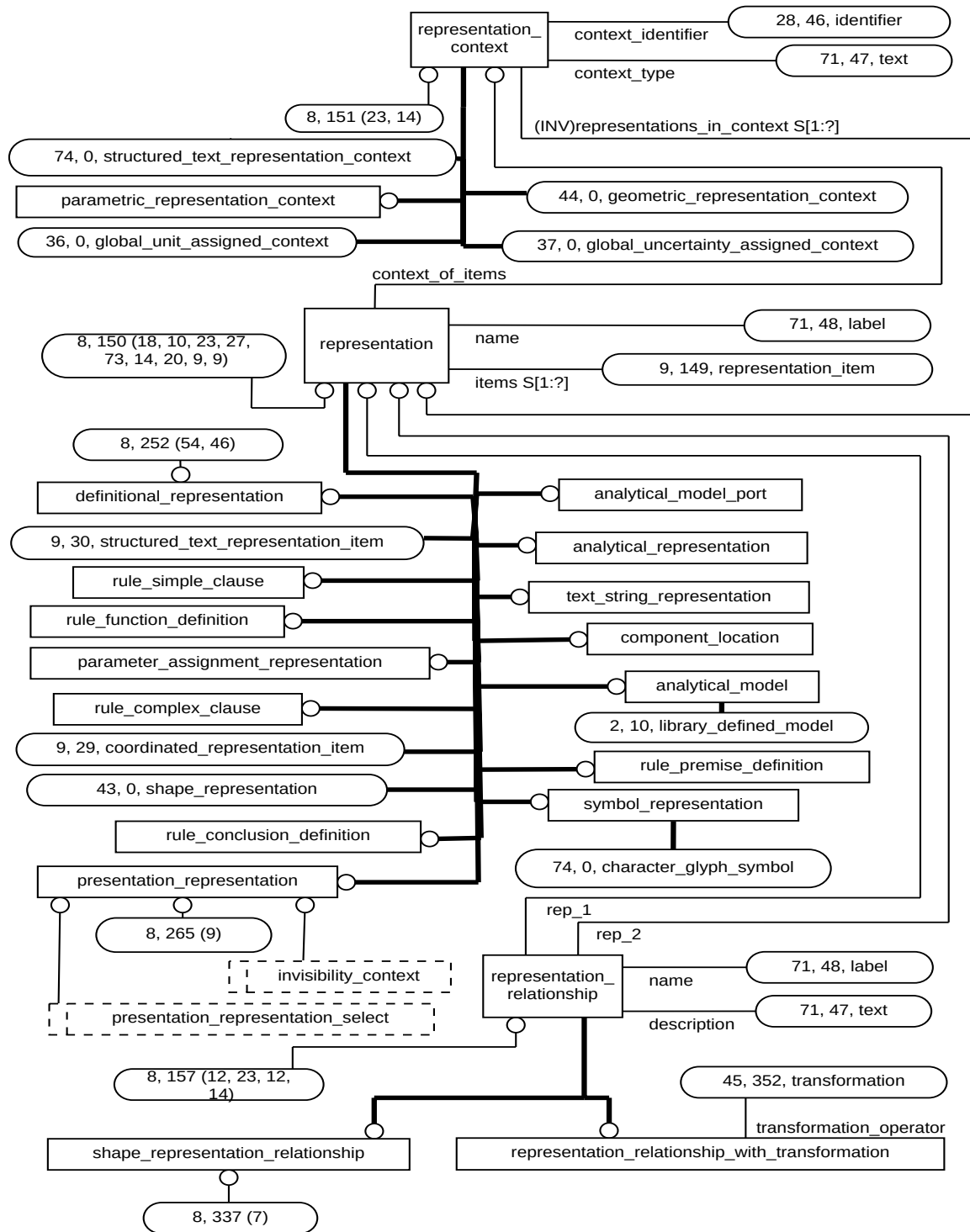


Figure J.8 – AIM EXPRESS-G diagram 8 of 80

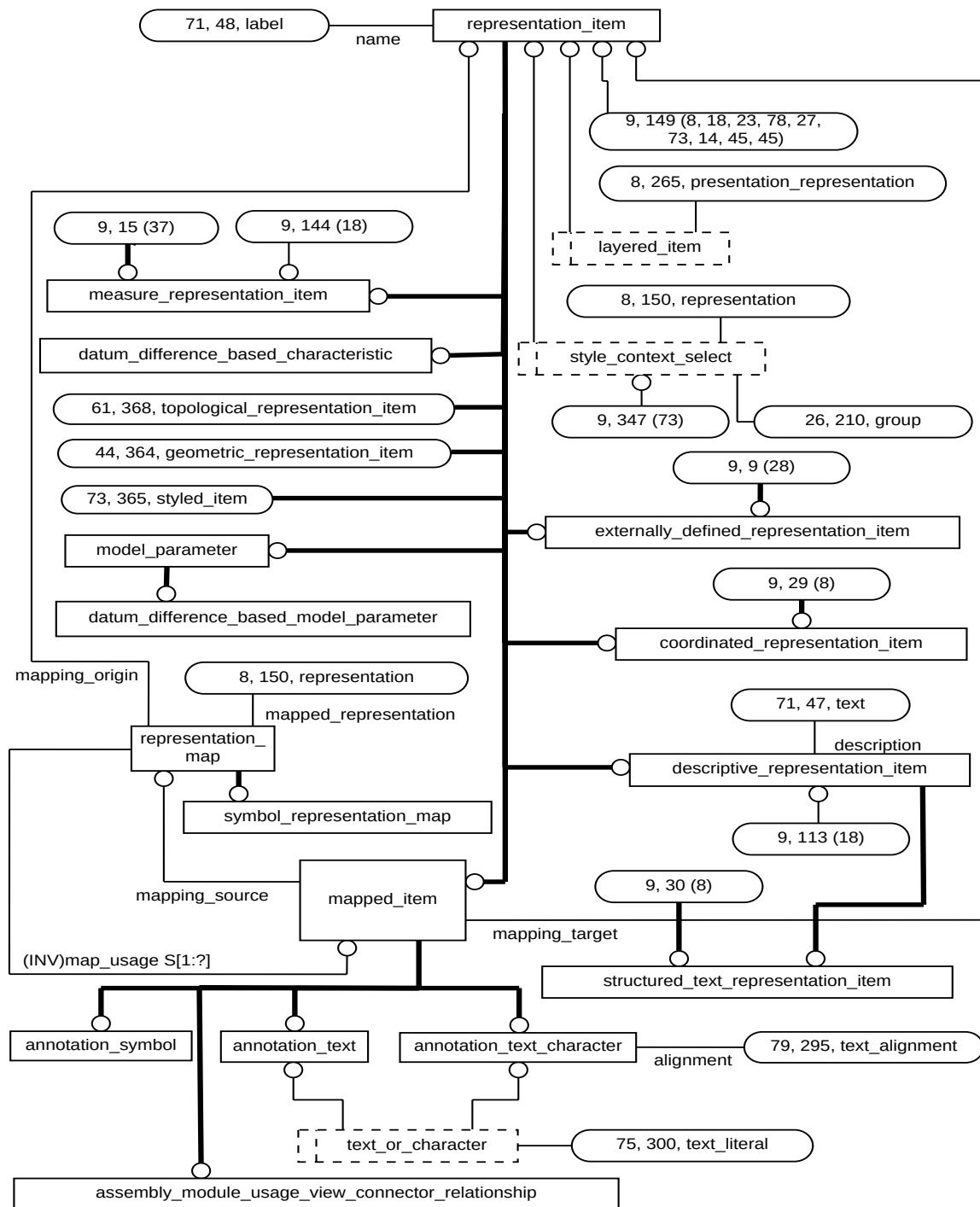


Figure J.9 – AIM EXPRESS-G diagram 9 of 80

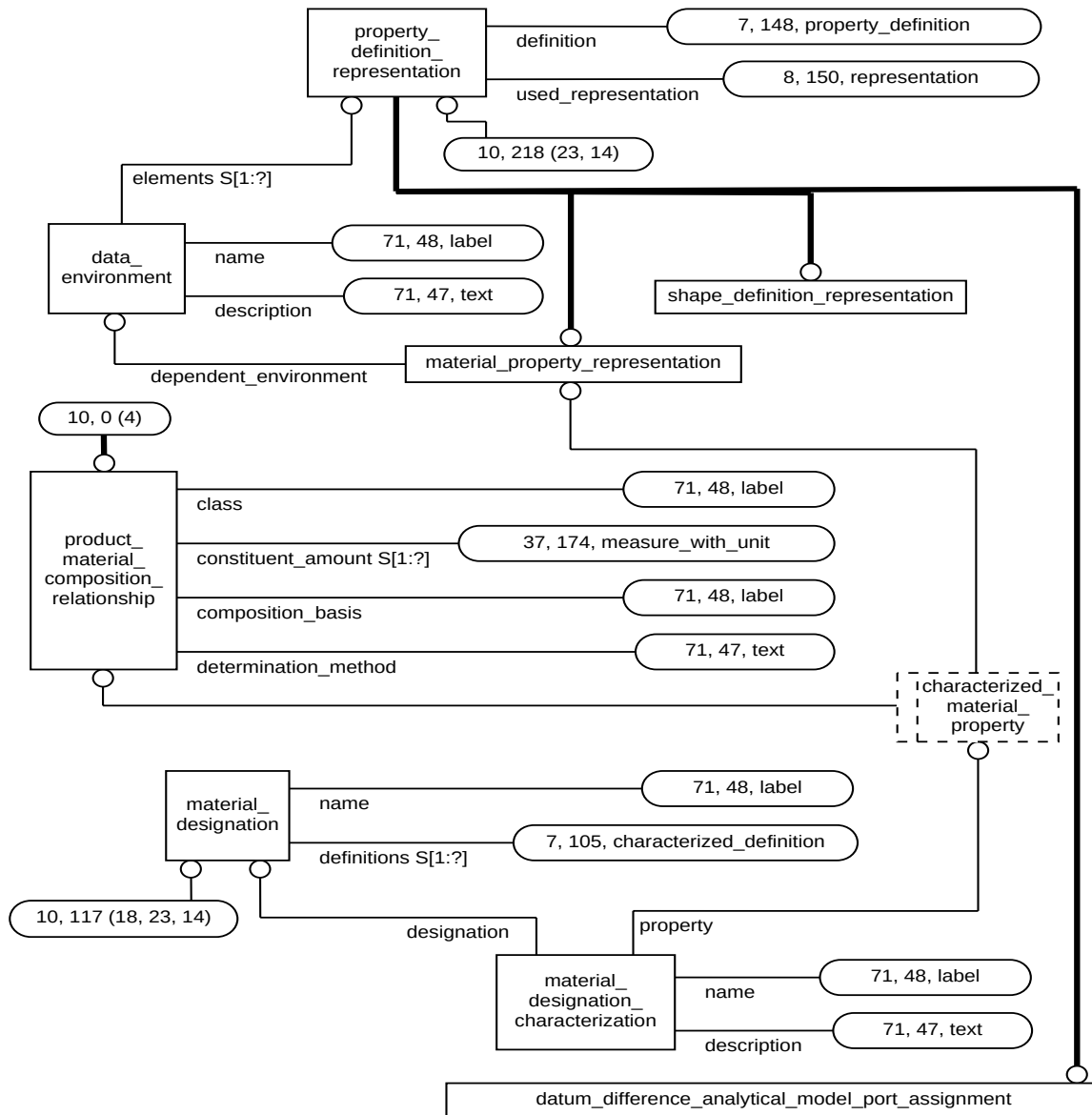


Figure J.10 – AIM EXPRESS-G diagram 10 of 80

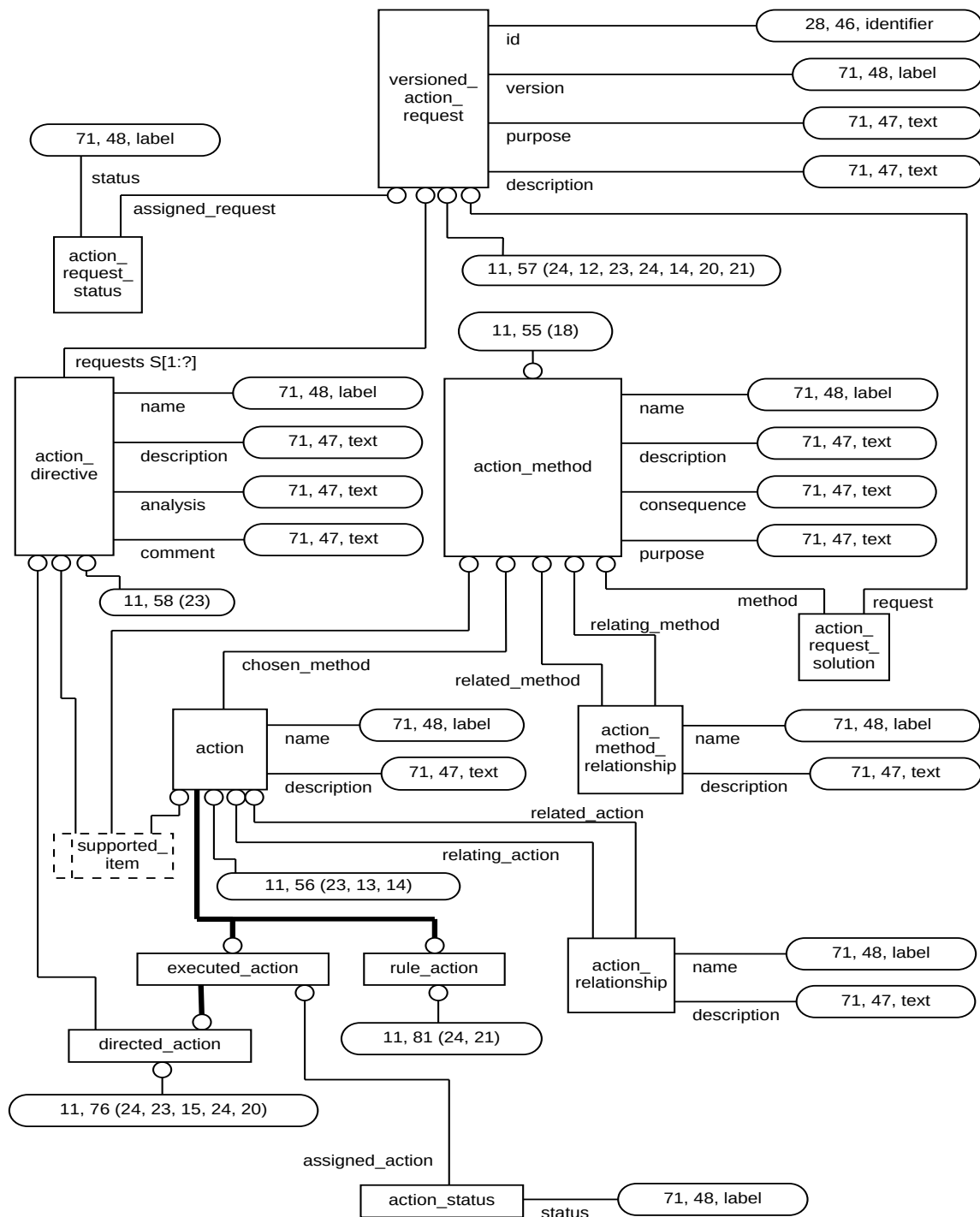


Figure J.11 – AIM EXPRESS-G diagram 11 of 80

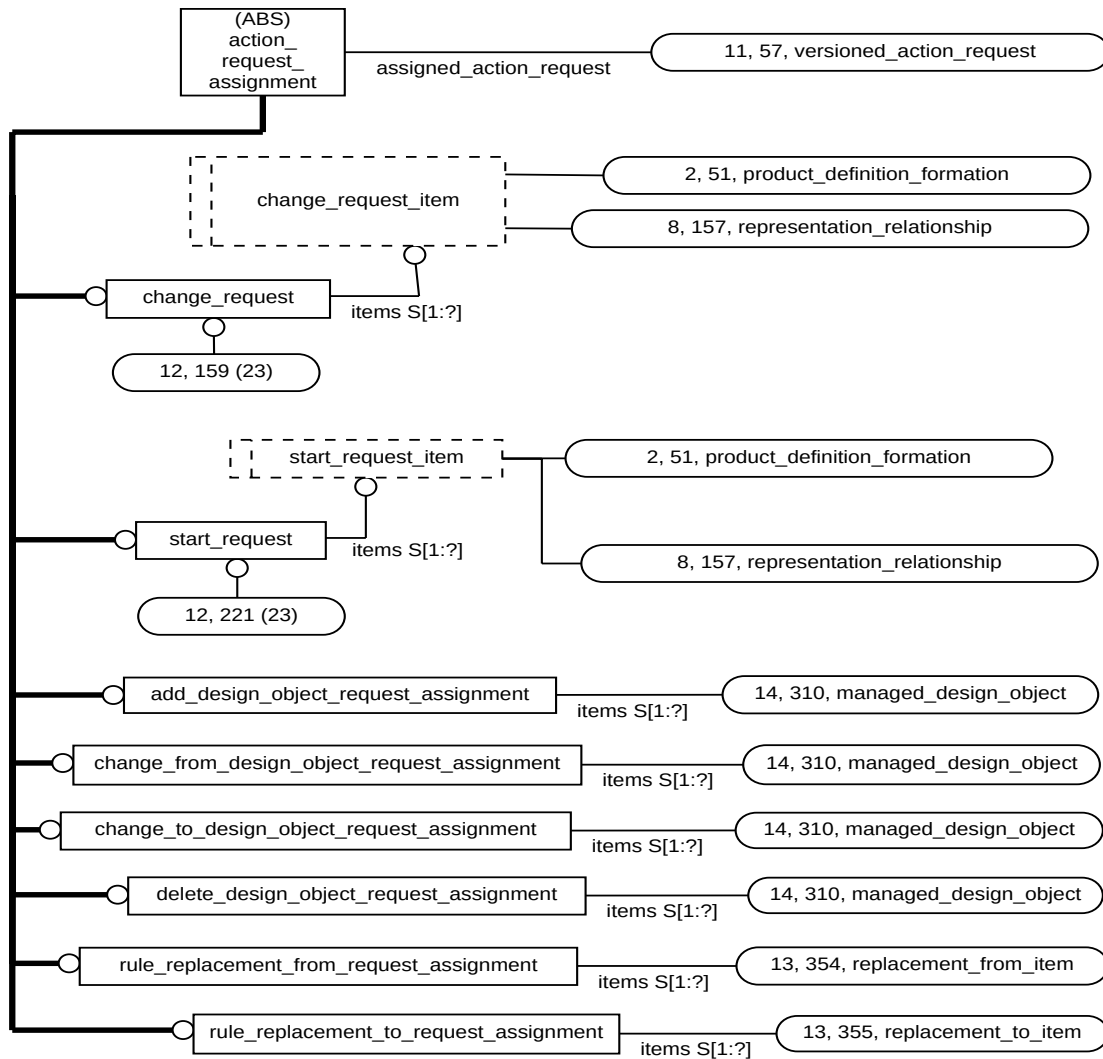


Figure J.12 – AIM EXPRESS-G diagram 12 of 80

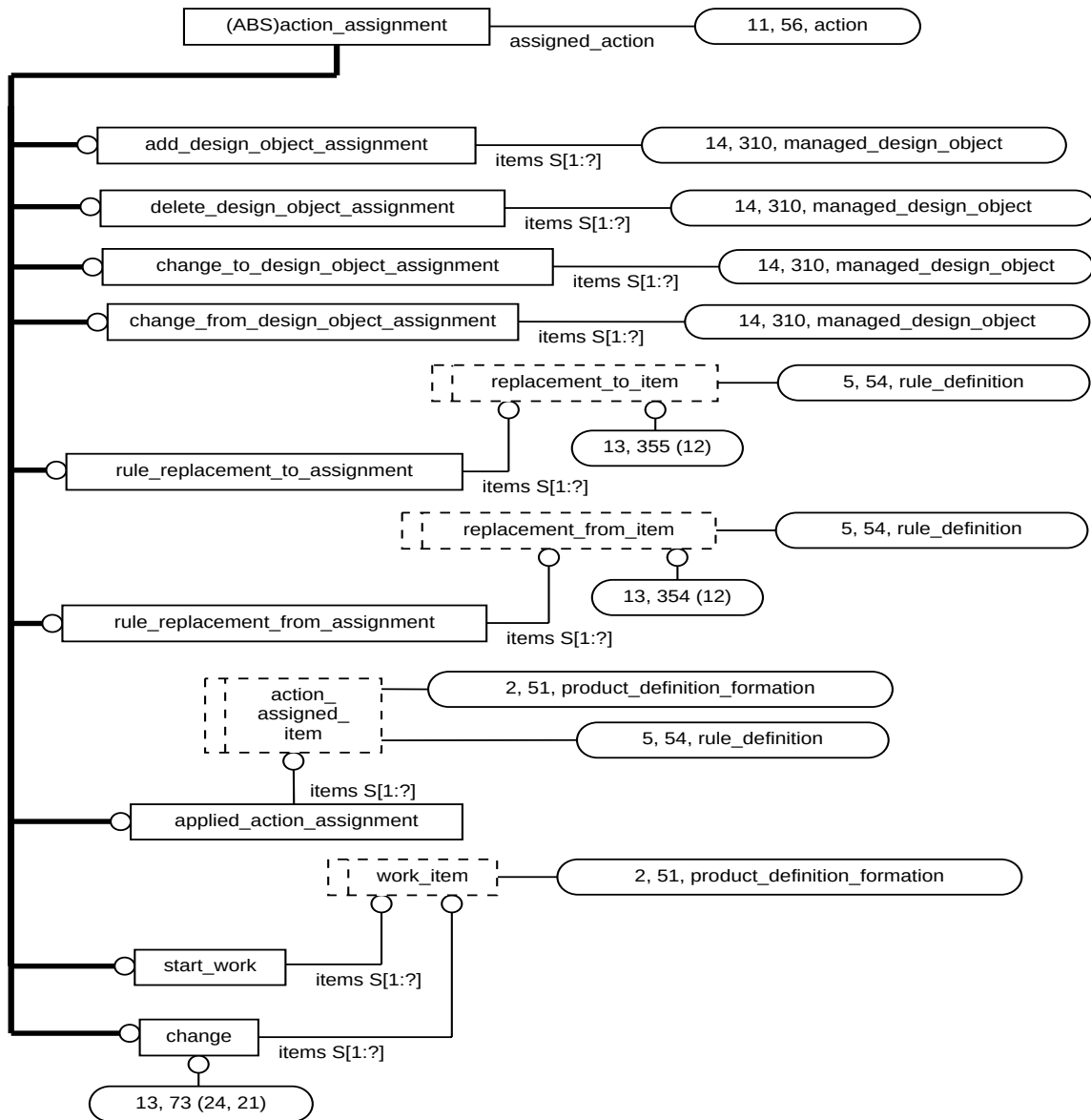


Figure J.13 – AIM EXPRESS-G diagram 13 of 80

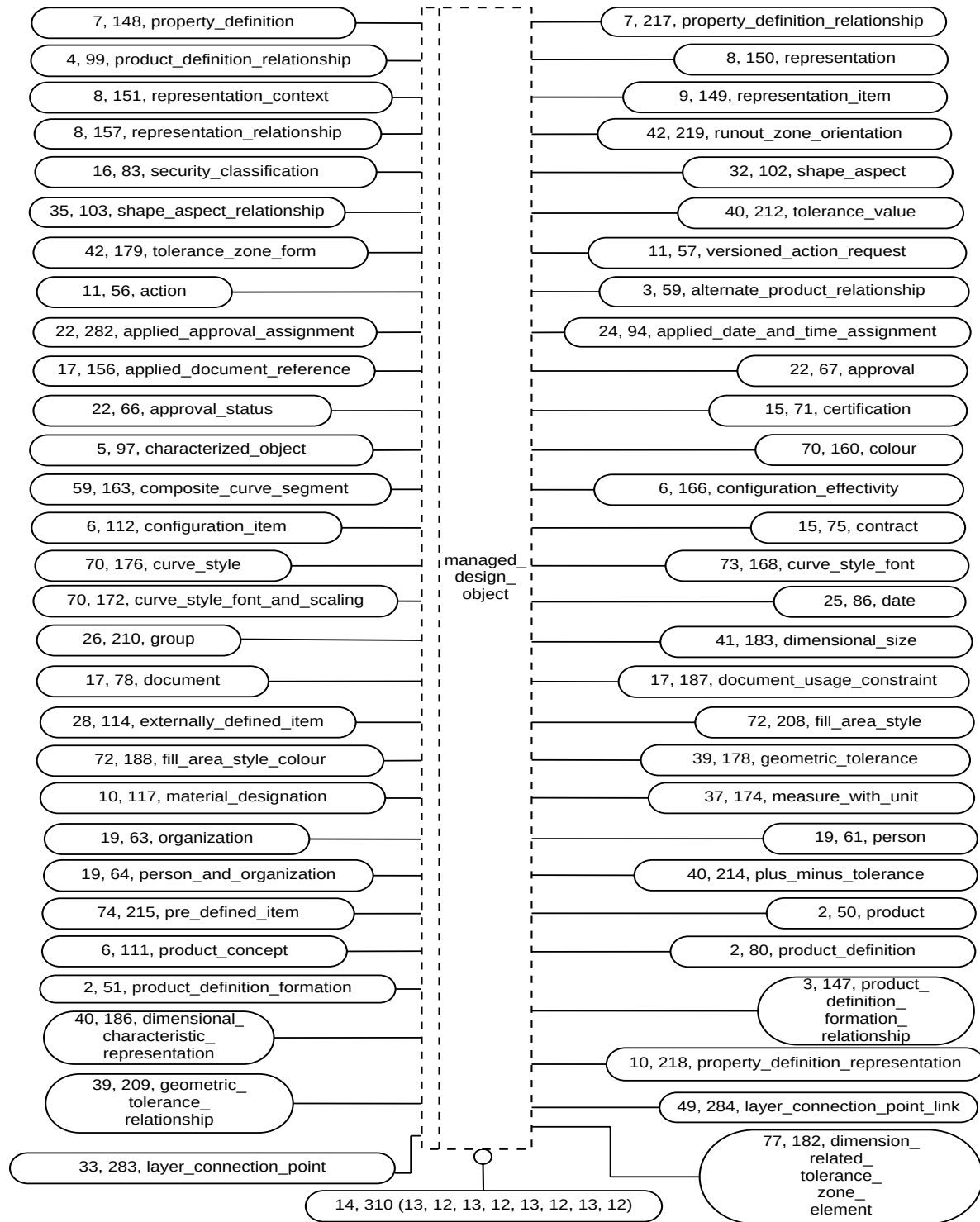


Figure J.14 – AIM EXPRESS-G diagram 14 of 80

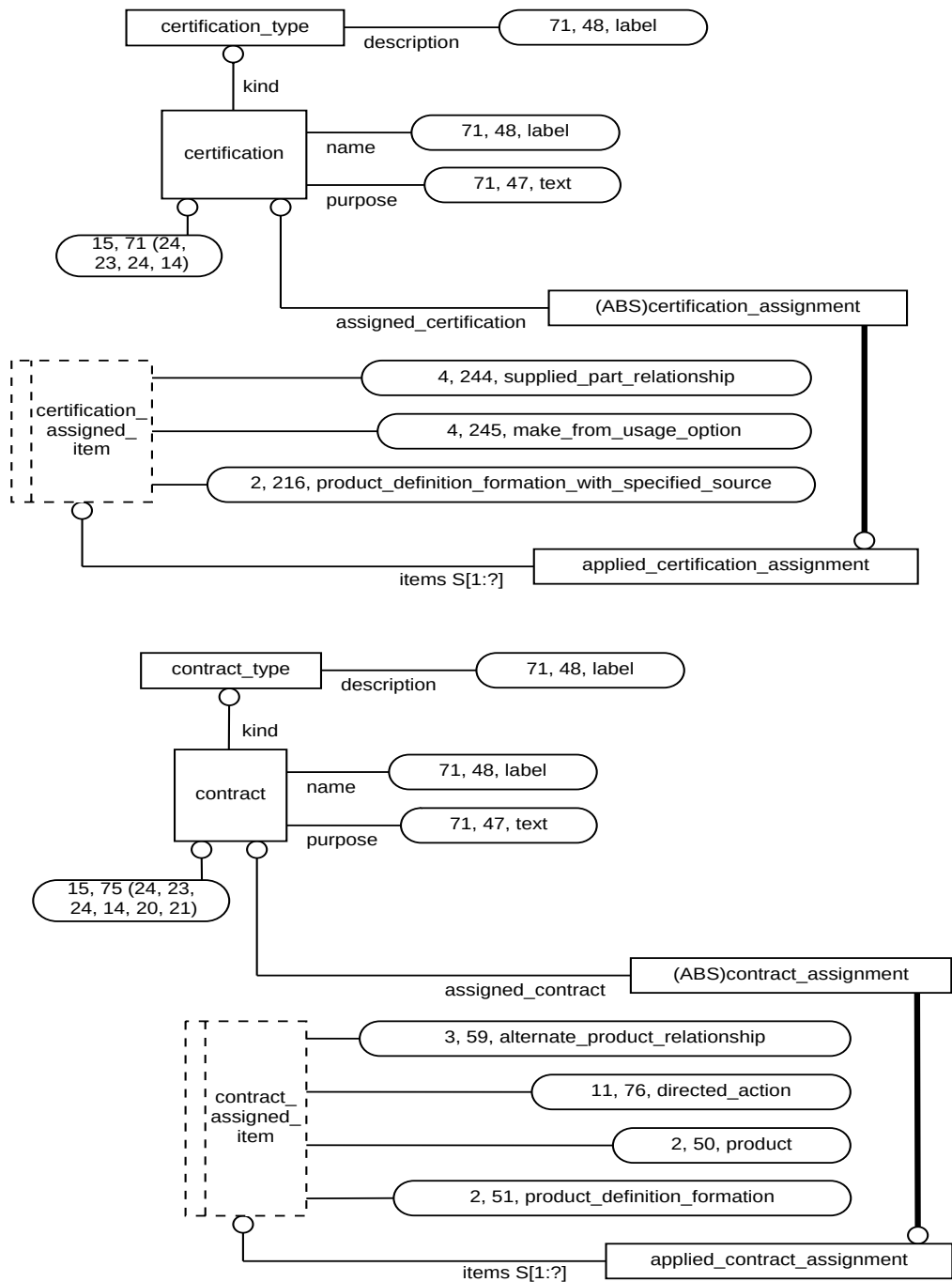


Figure J.15 – AIM EXPRESS-G diagram 15 of 80

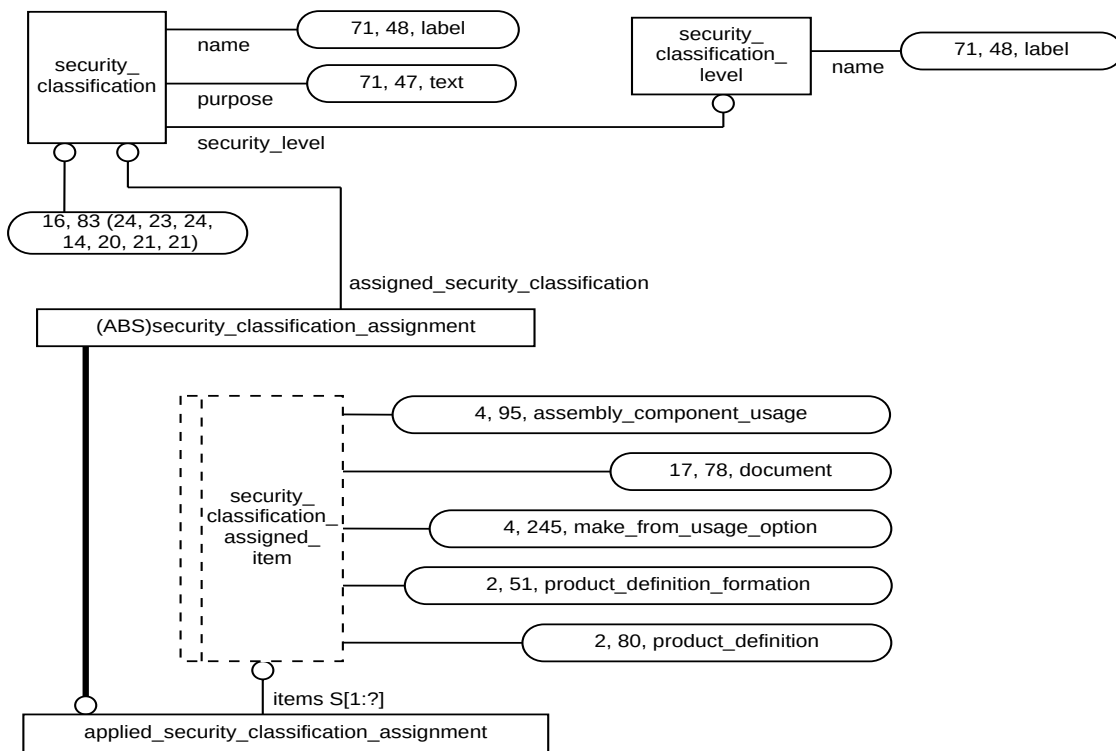


Figure J.16 – AIM EXPRESS-G diagram 16 of 80

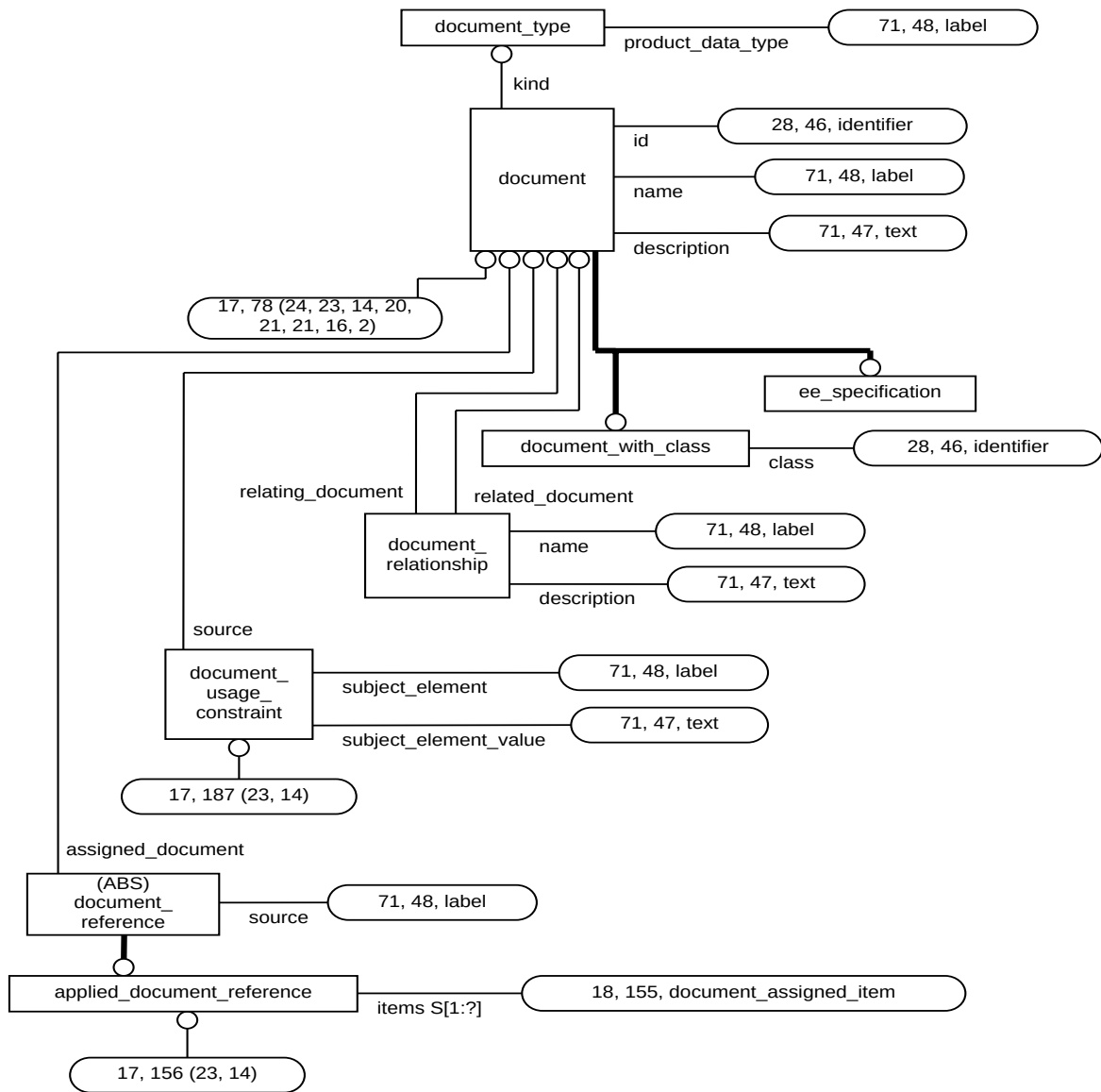


Figure J.17 – AIM EXPRESS-G diagram 17 of 80

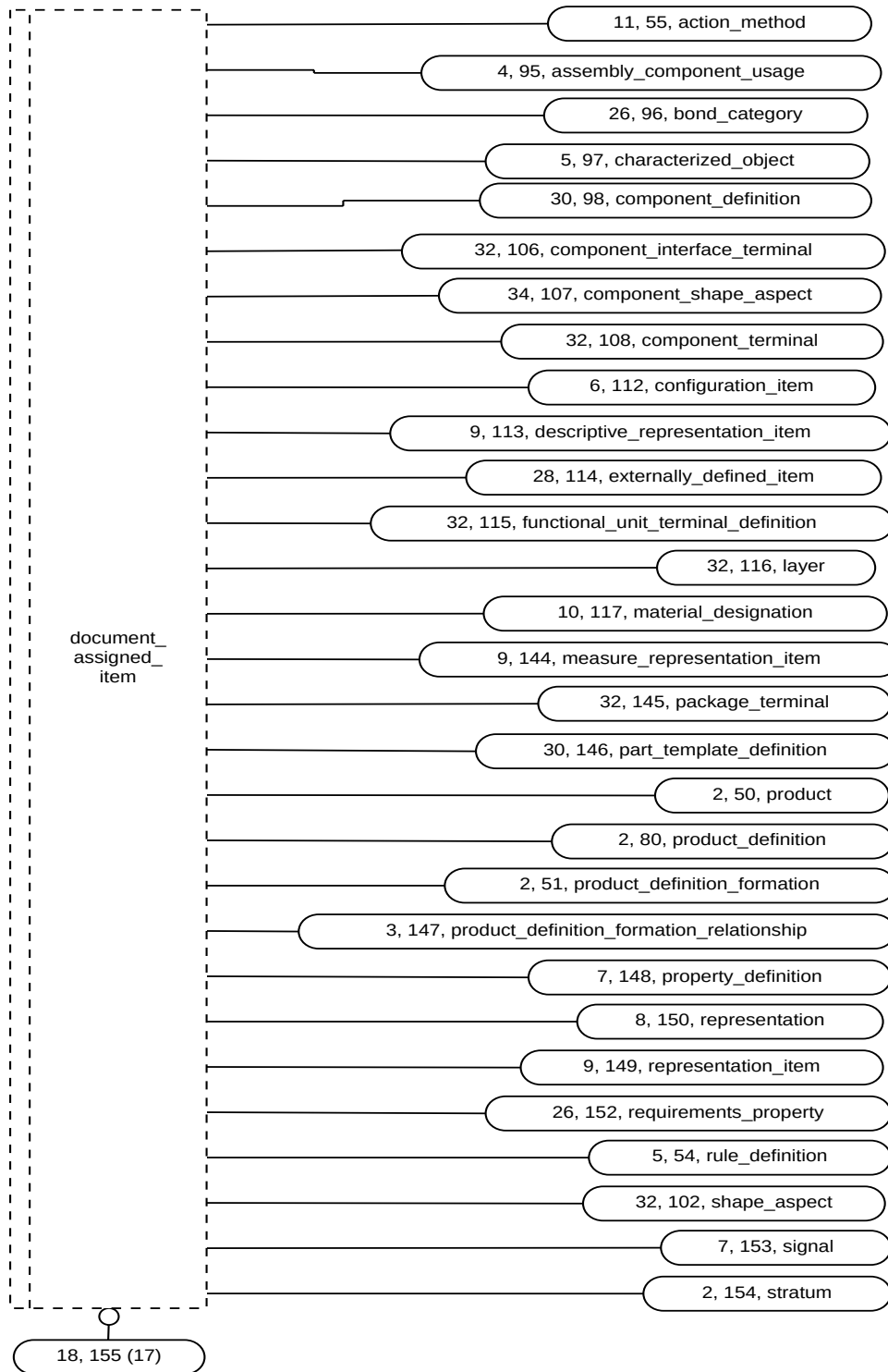


Figure J.18 – AIM EXPRESS-G diagram 18 of 80

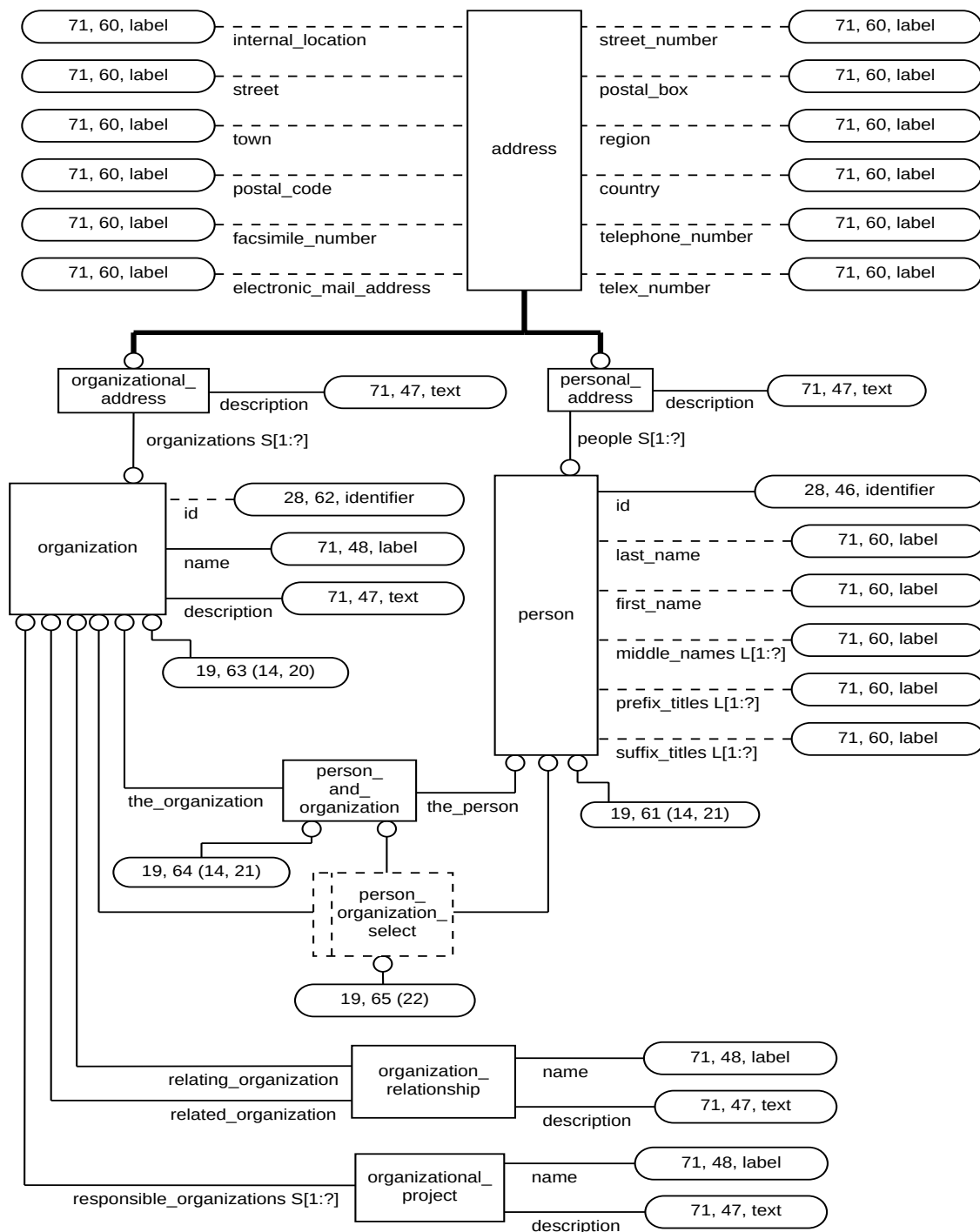


Figure J.19 – AIM EXPRESS-G diagram 19 of 80

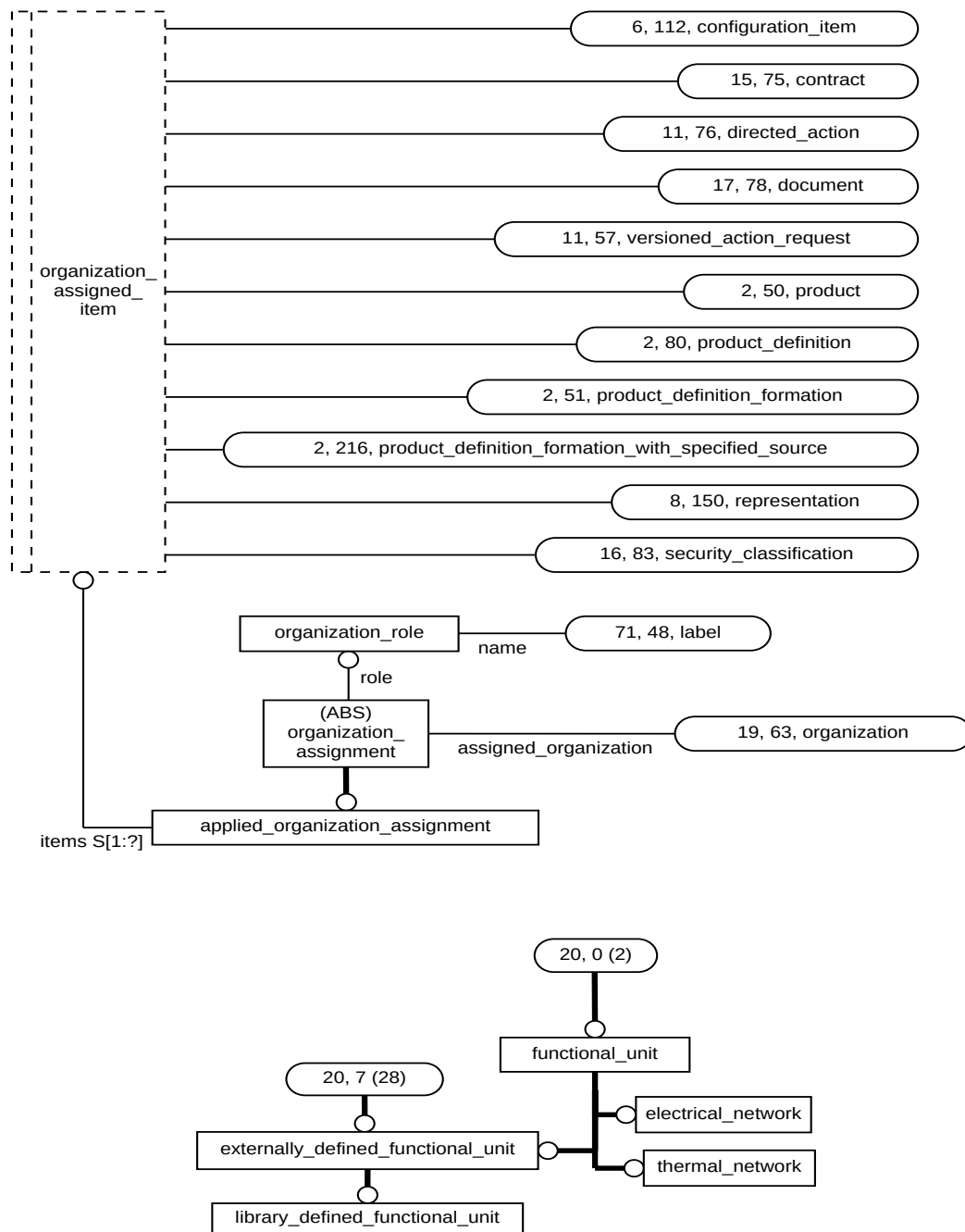


Figure J.20 – AIM EXPRESS-G diagram 20 of 80

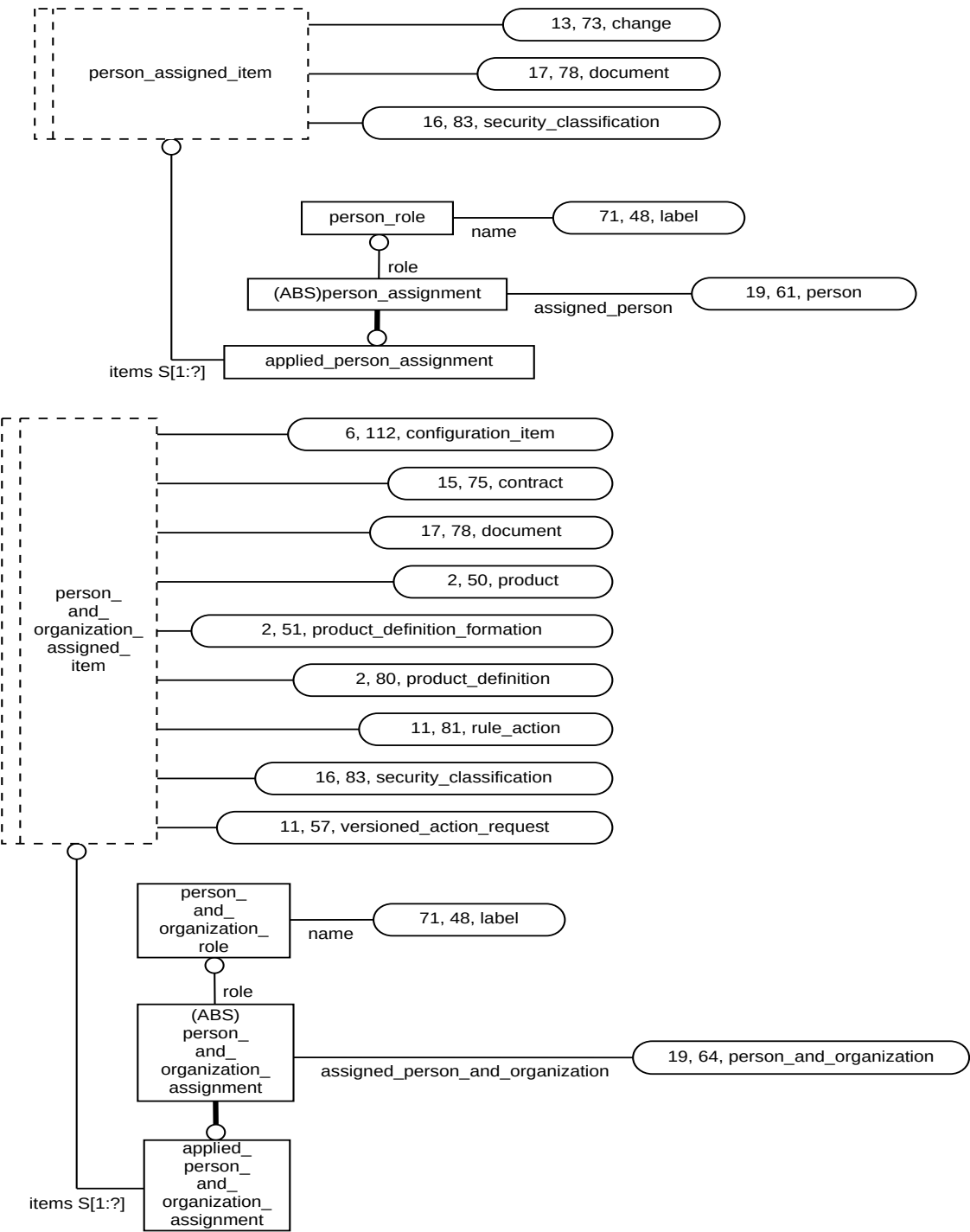


Figure J.21 – AIM EXPRESS-G diagram 21 of 80

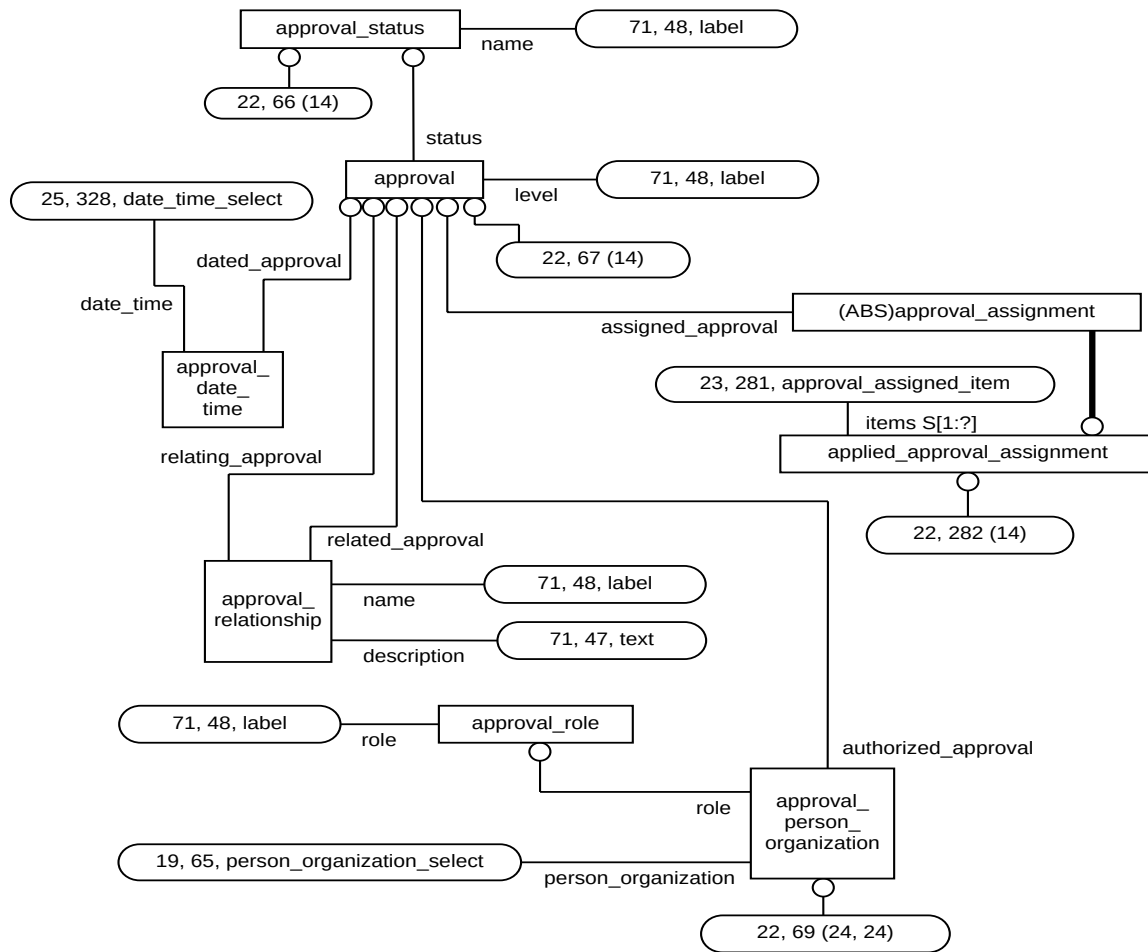


Figure J.22 – AIM EXPRESS-G diagram 22 of 80

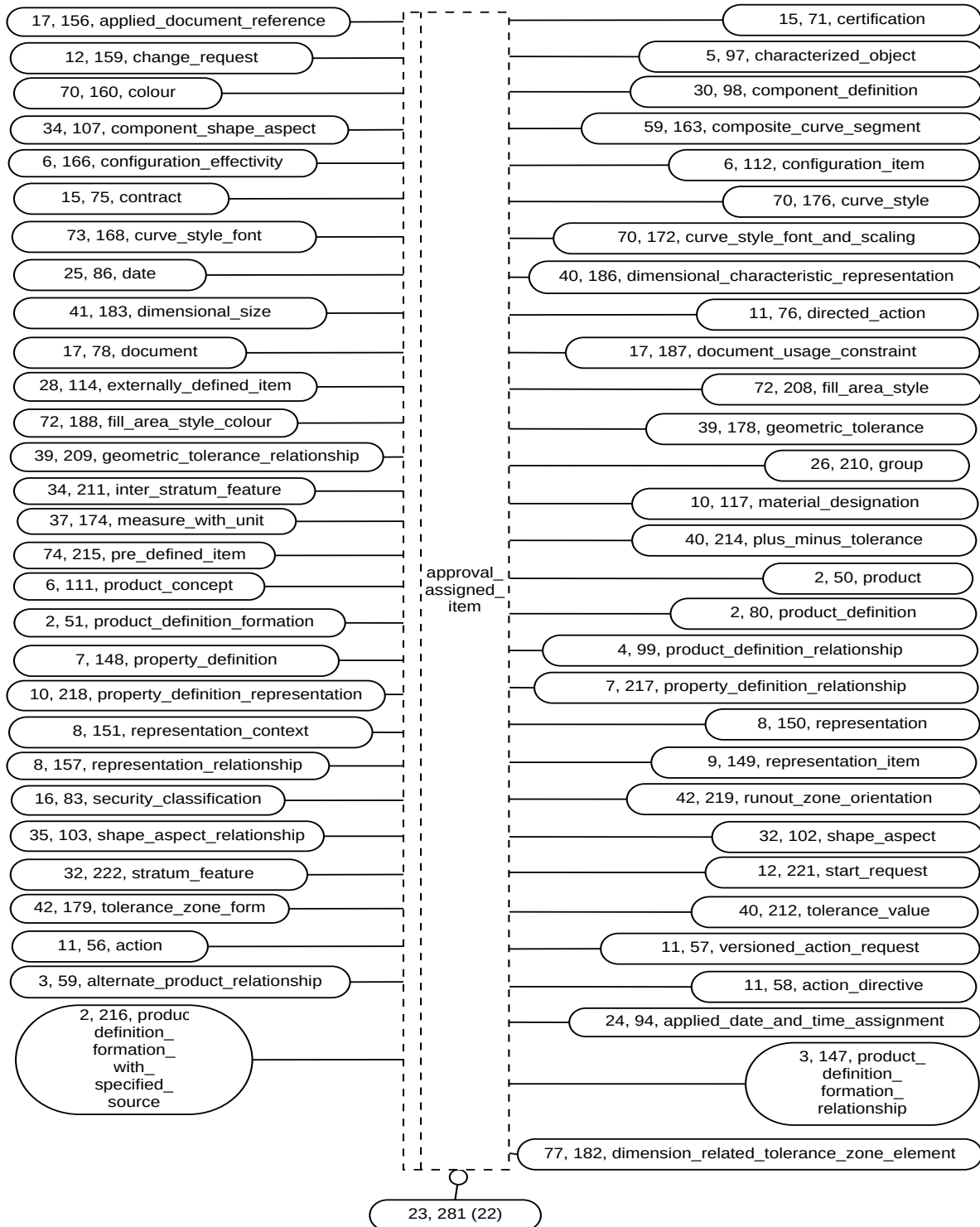


Figure J.23 – AIM EXPRESS-G diagram 23 of 80

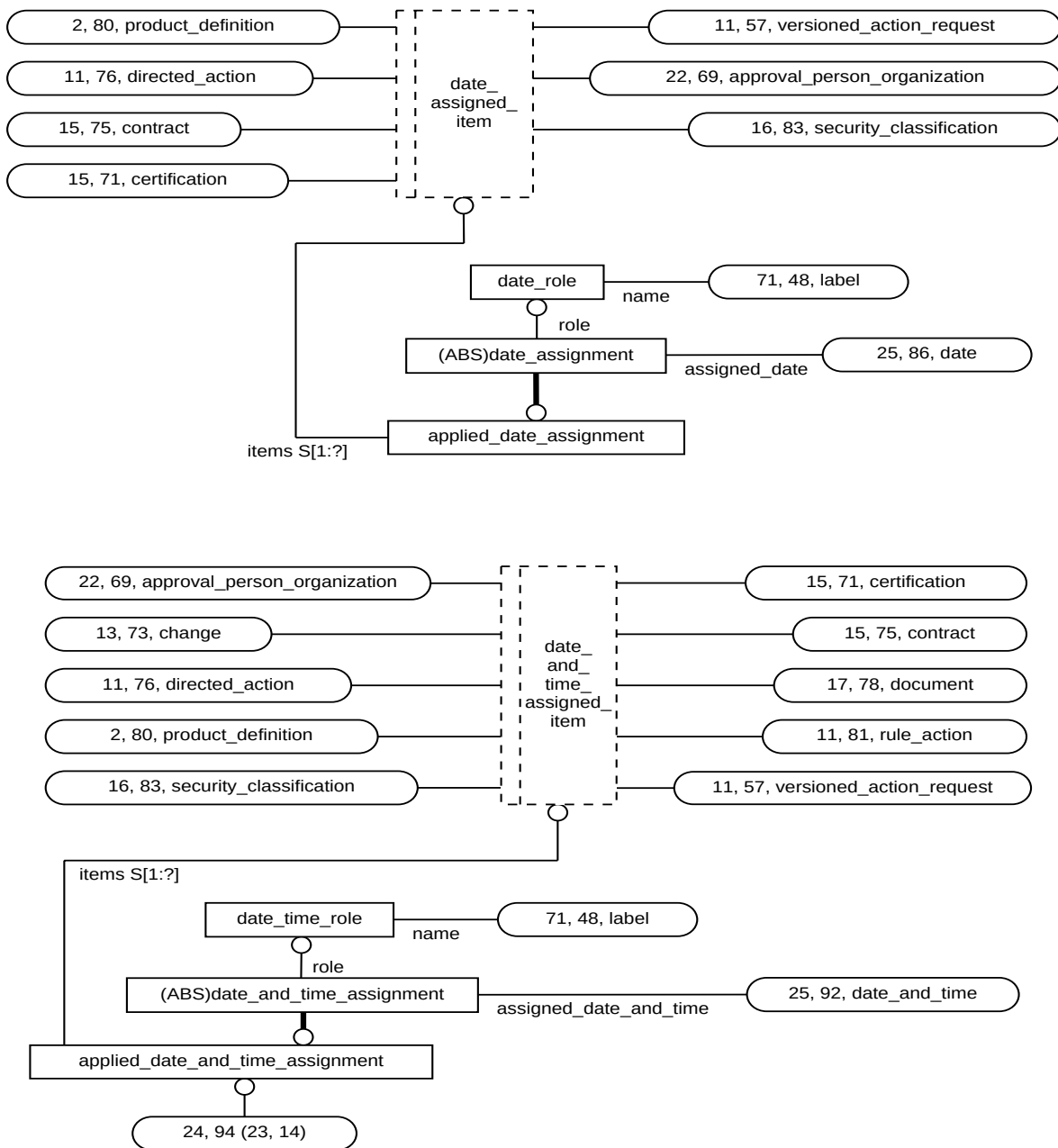


Figure J.24 – AIM EXPRESS-G diagram 24 of 80

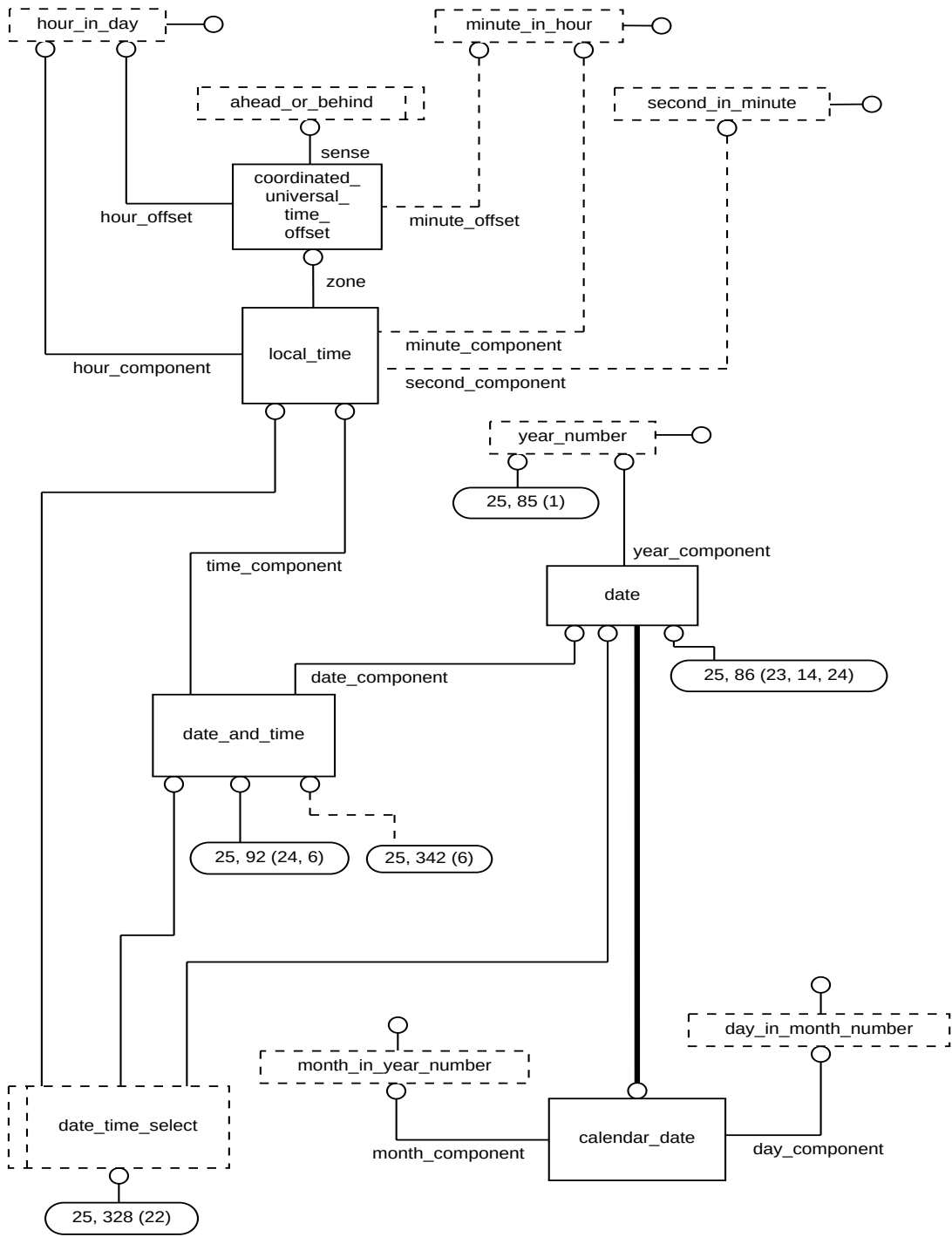


Figure J.25 – AIM EXPRESS-G diagram 25 of 80

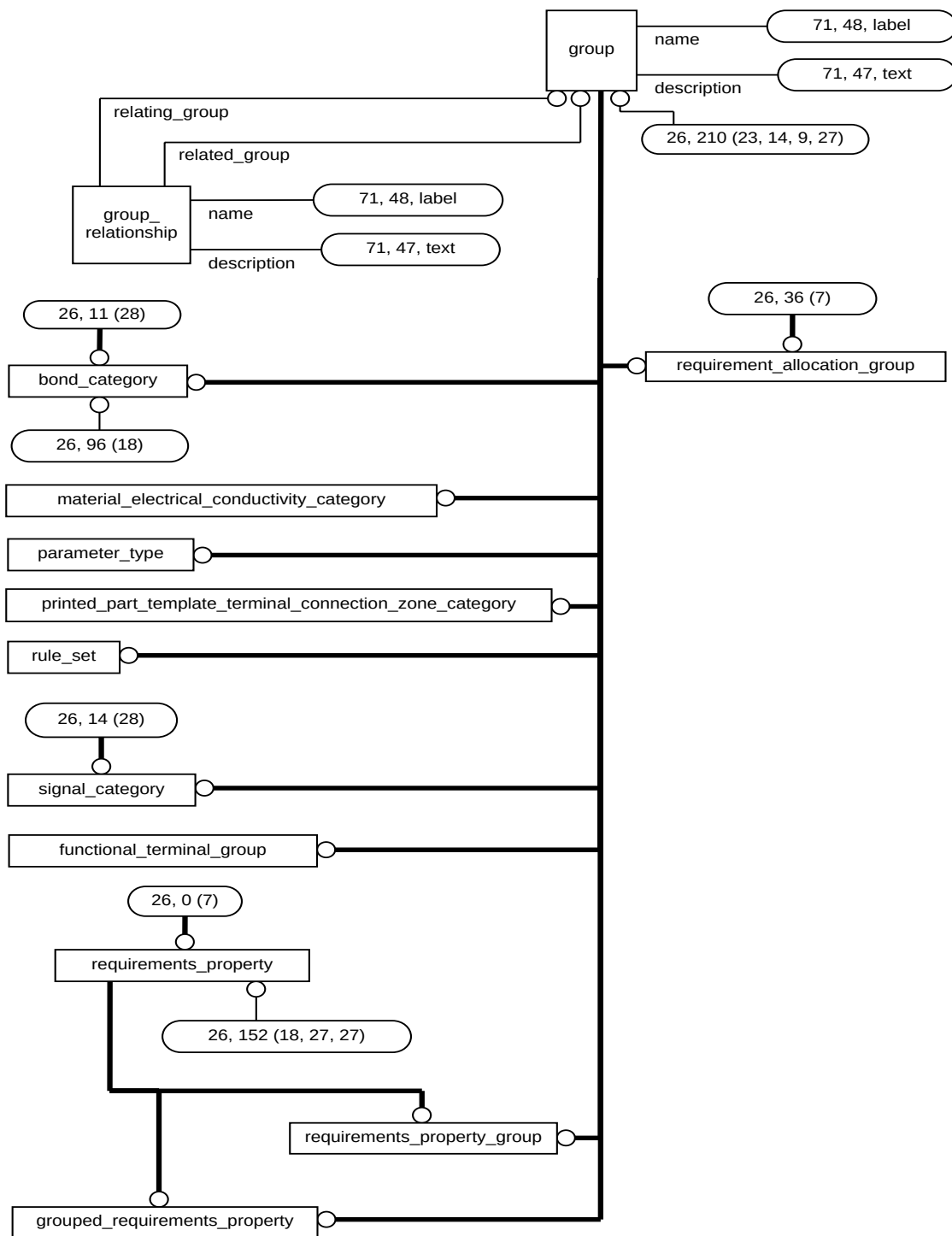


Figure J.26 – AIM EXPRESS-G diagram 26 of 80

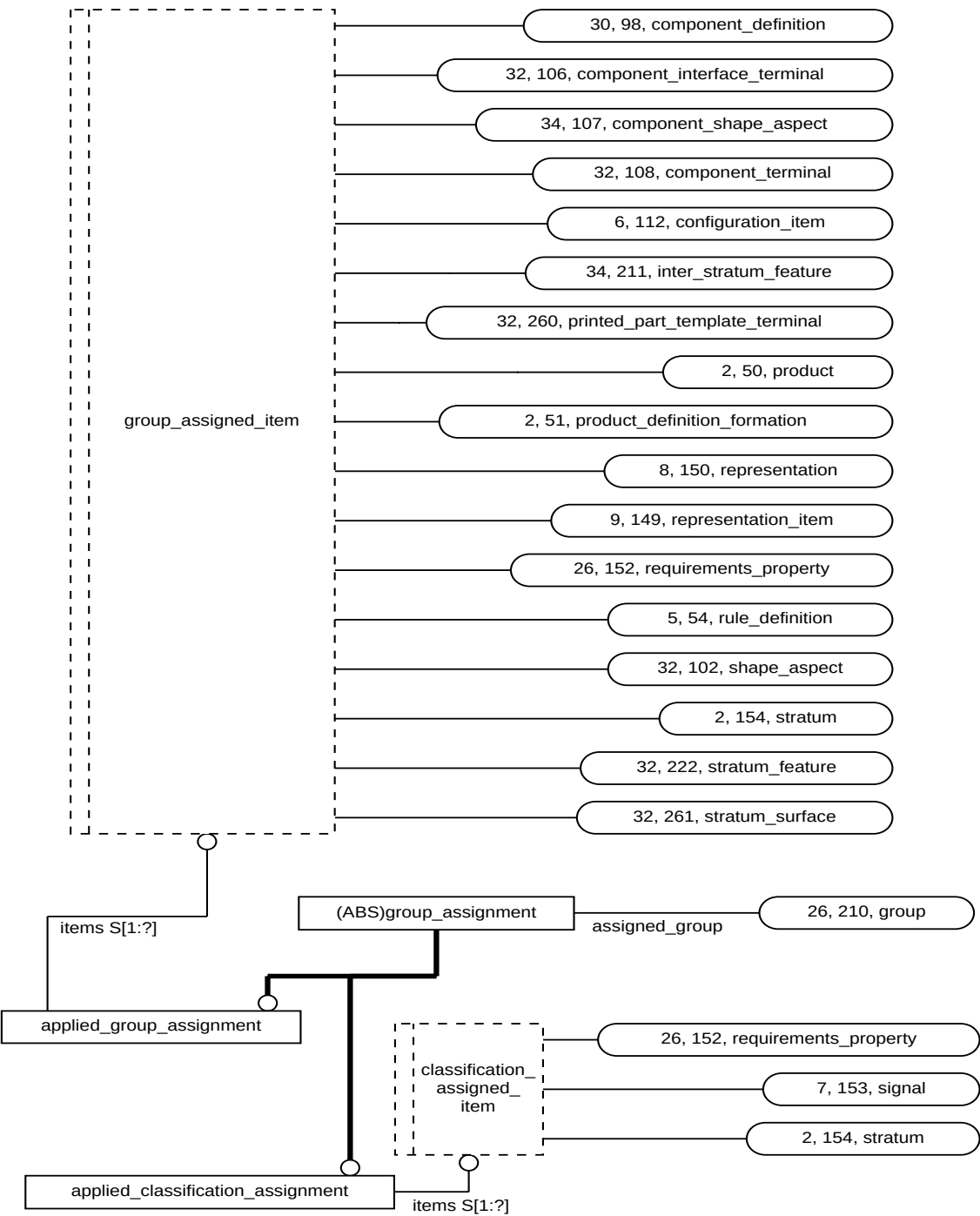


Figure J.27 – AIM EXPRESS-G diagram 27 of 80

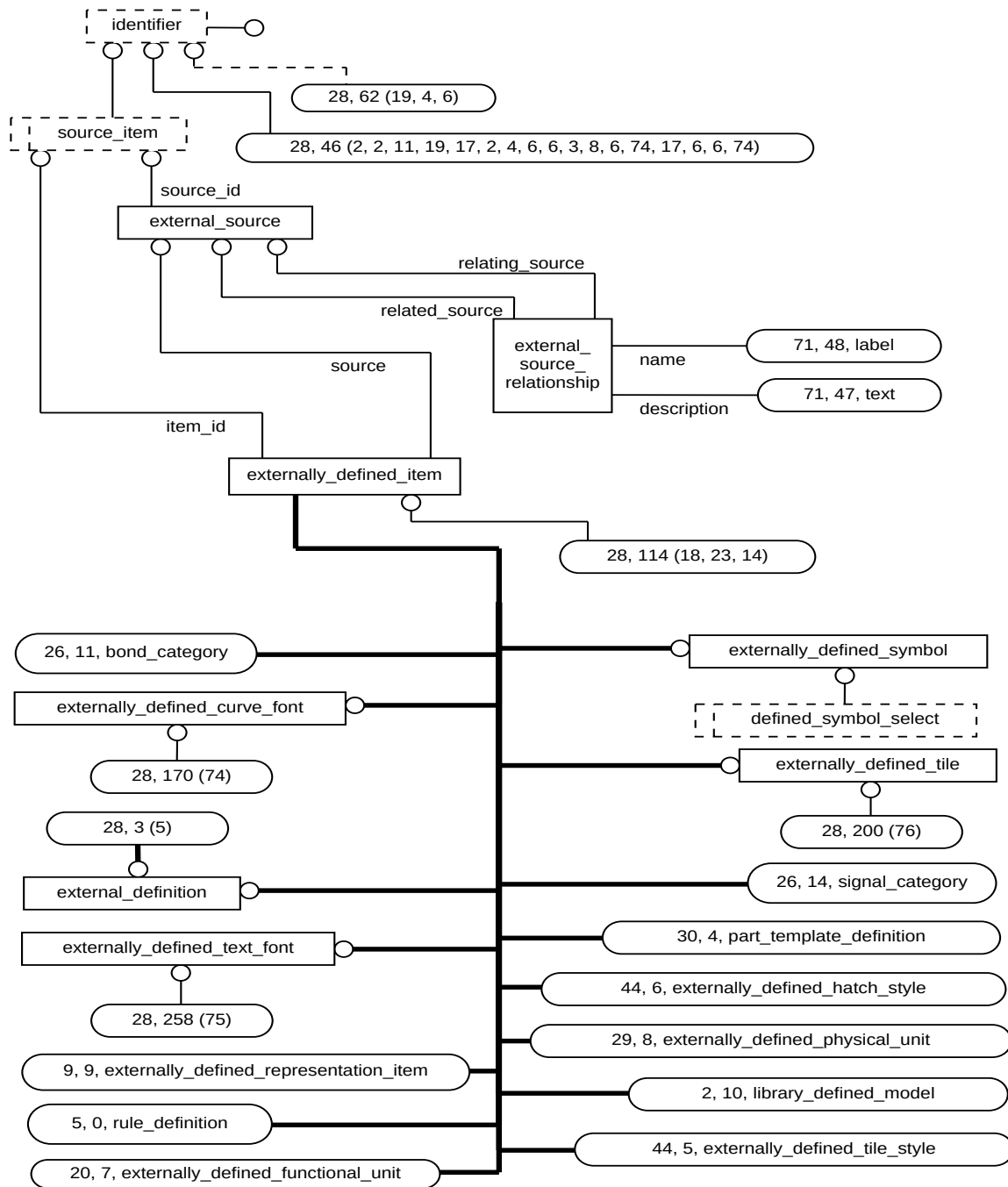


Figure J.28 – AIM EXPRESS-G diagram 28 of 80

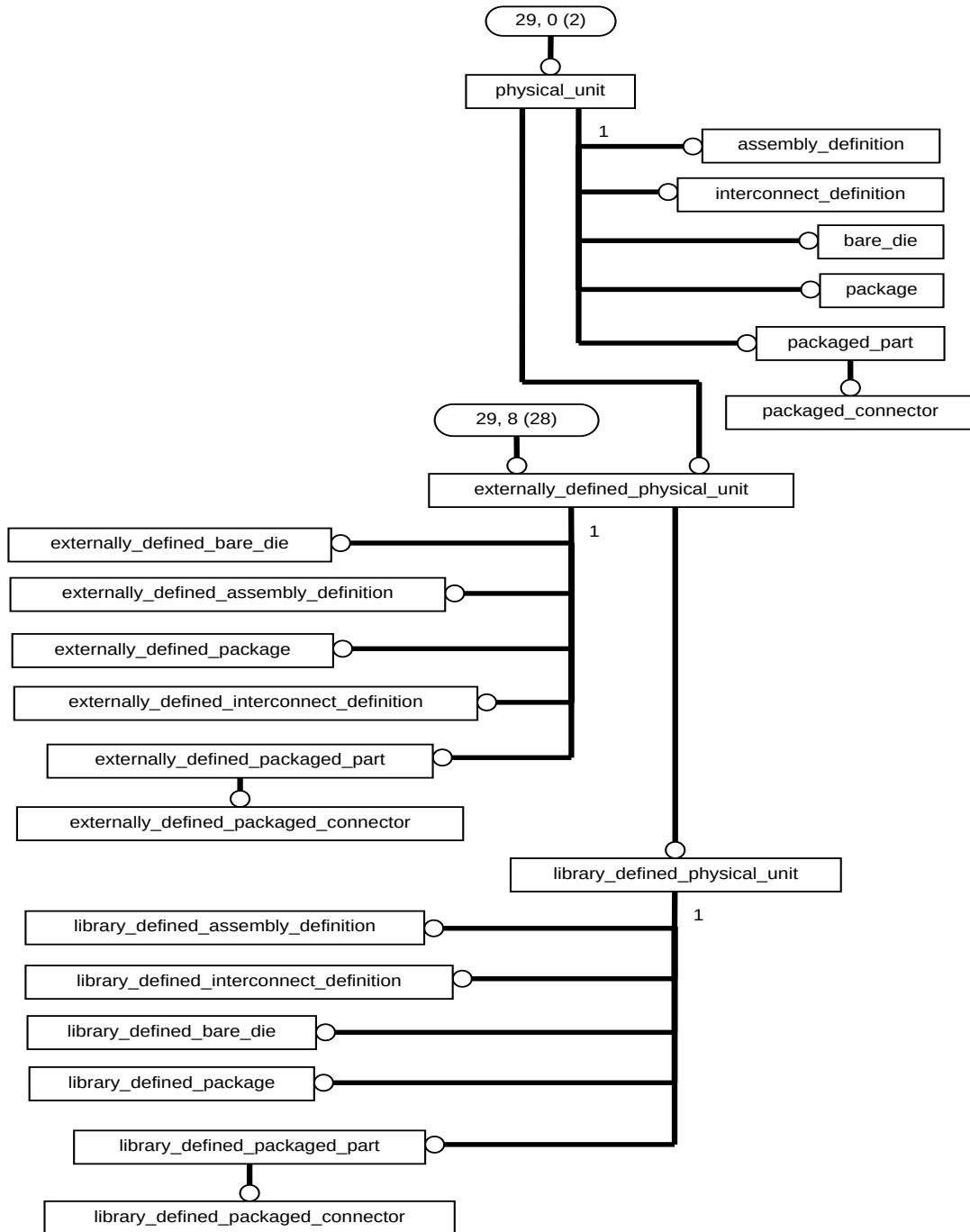


Figure J.29 – AIM EXPRESS-G diagram 29 of 80

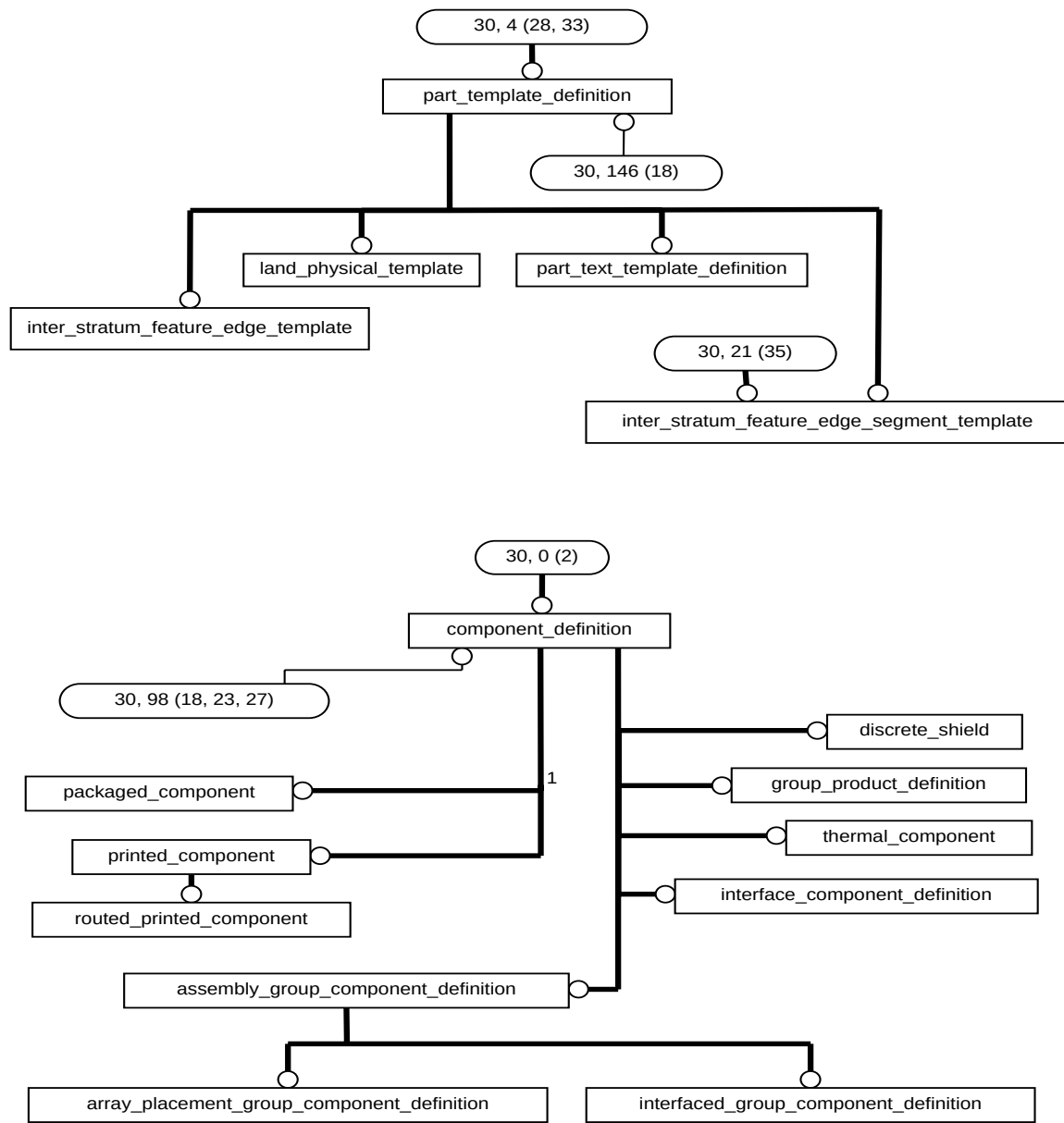


Figure J.30 – AIM EXPRESS-G diagram 30 of 80

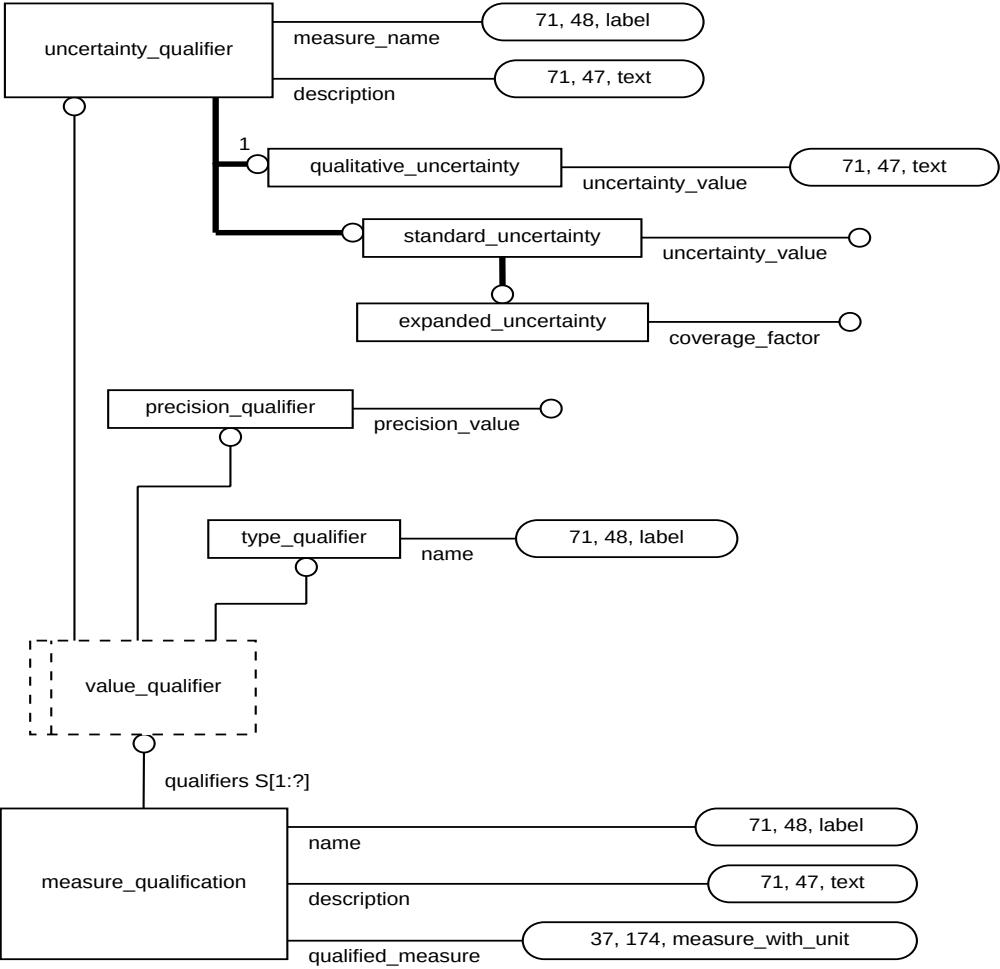


Figure J.31 – AIM EXPRESS-G diagram 31 of 80

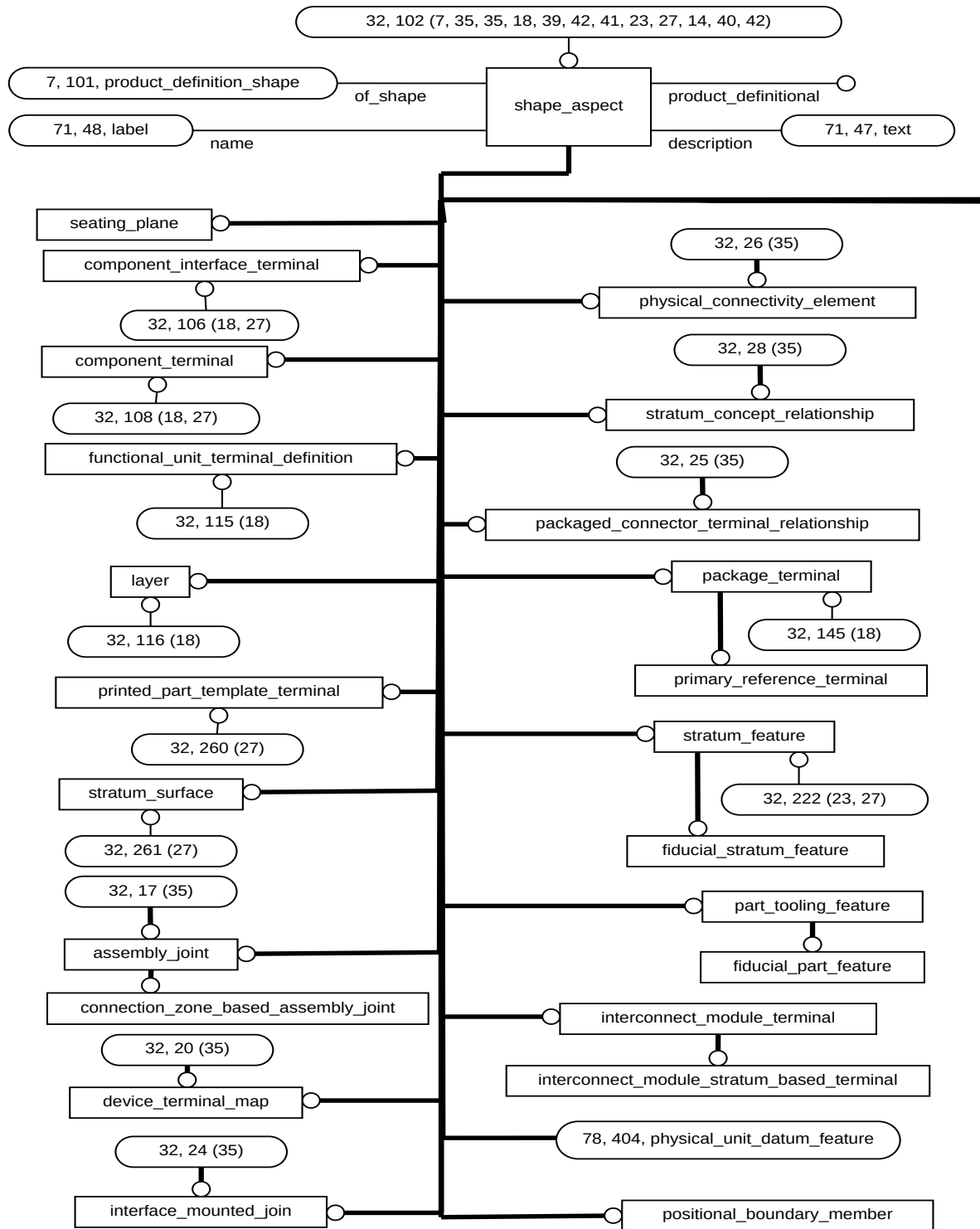


Figure J.32 – AIM EXPRESS-G diagram 32 of 80

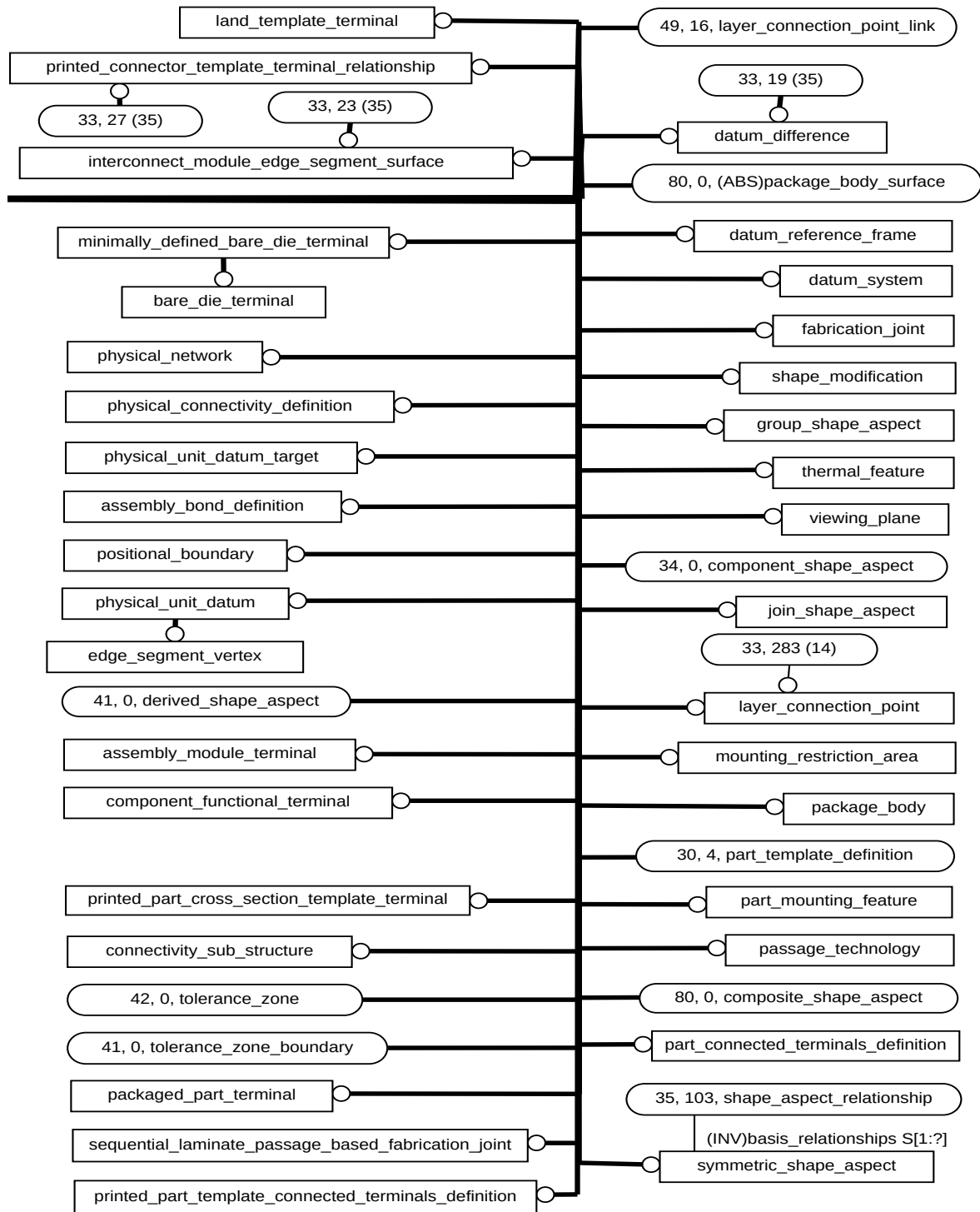


Figure J.33 – AIM EXPRESS-G diagram 33 of 80

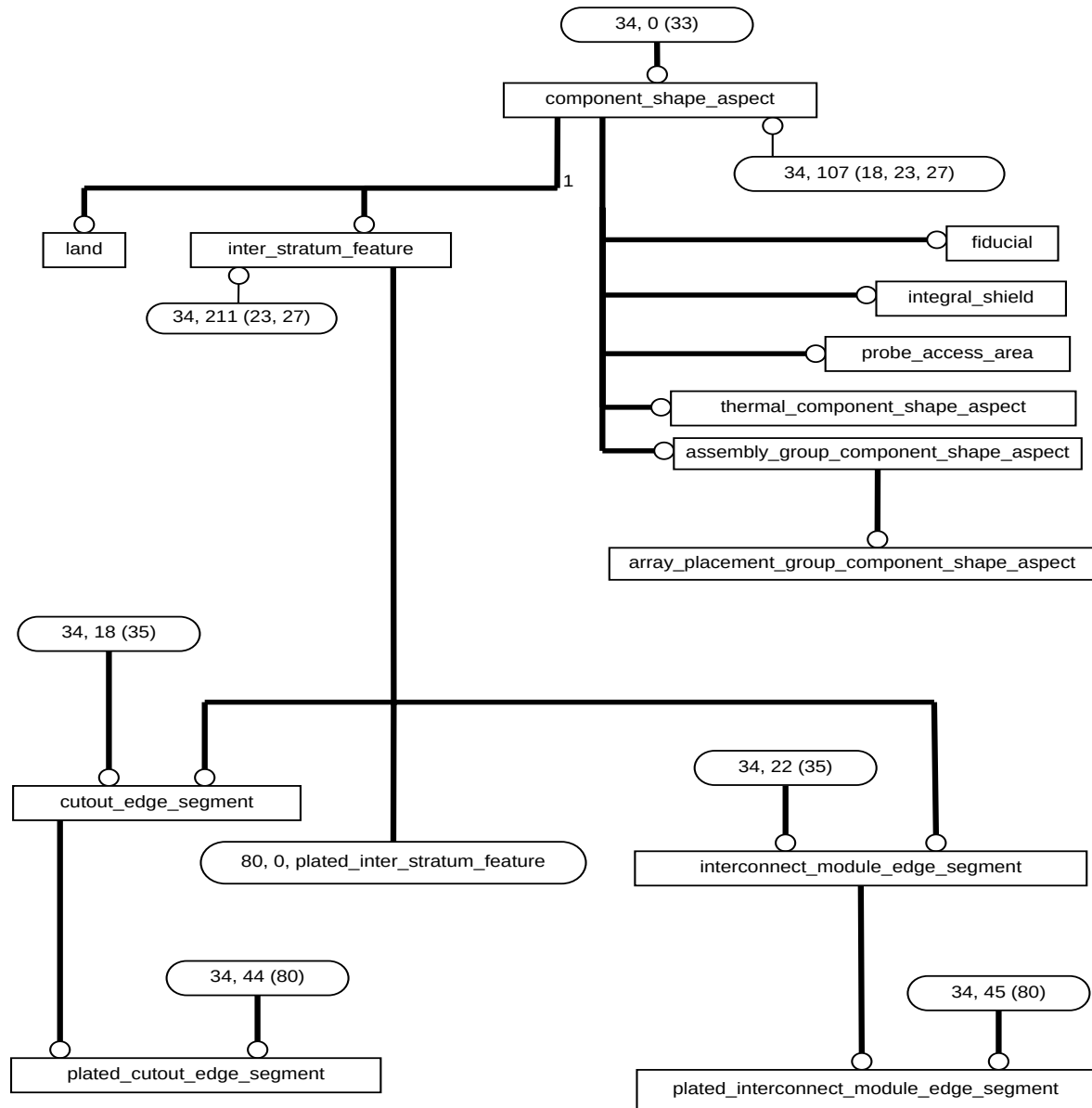


Figure J.34 – AIM EXPRESS-G diagram 34 of 80

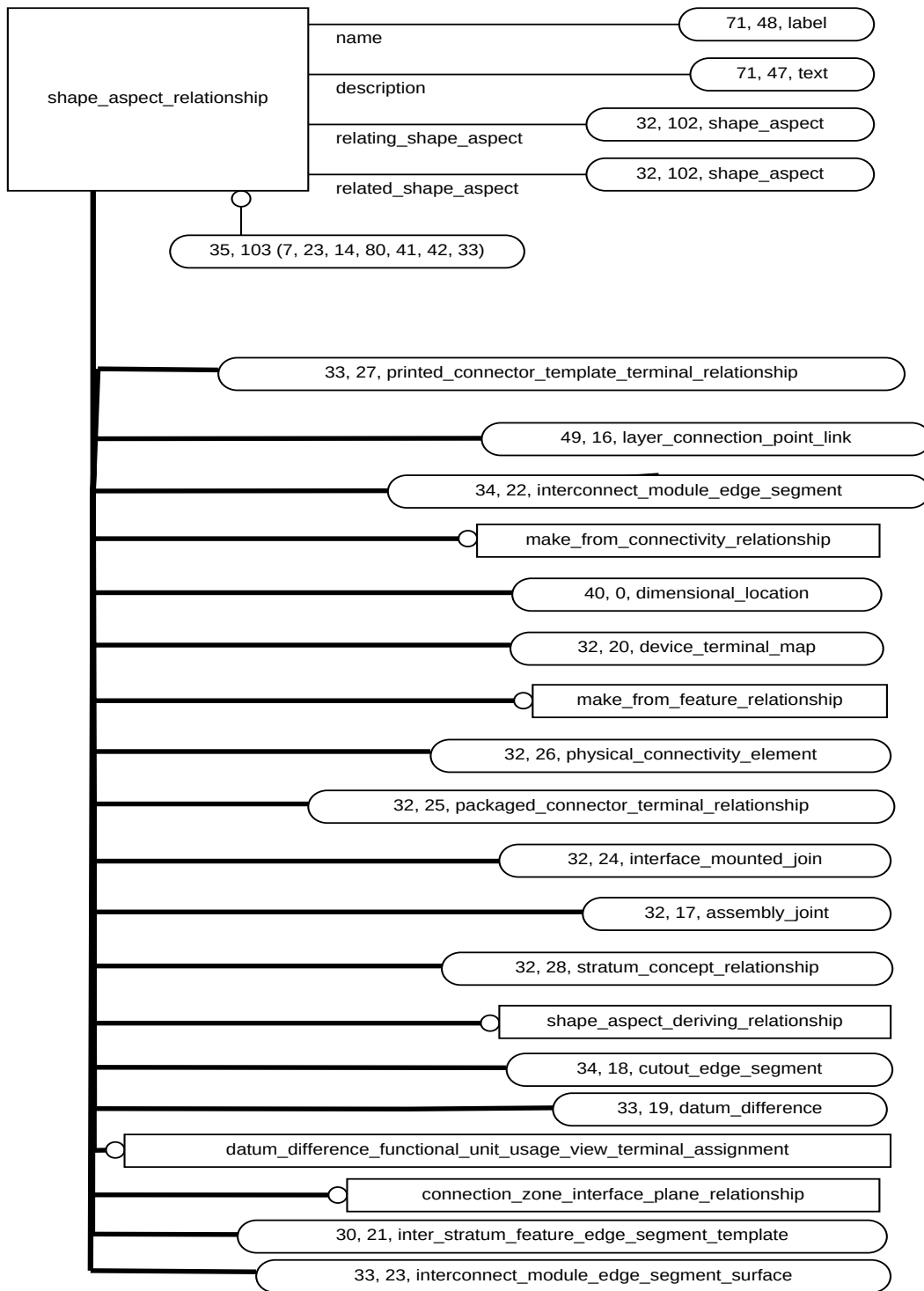


Figure J.35 – AIM EXPRESS-G diagram 35 of 80

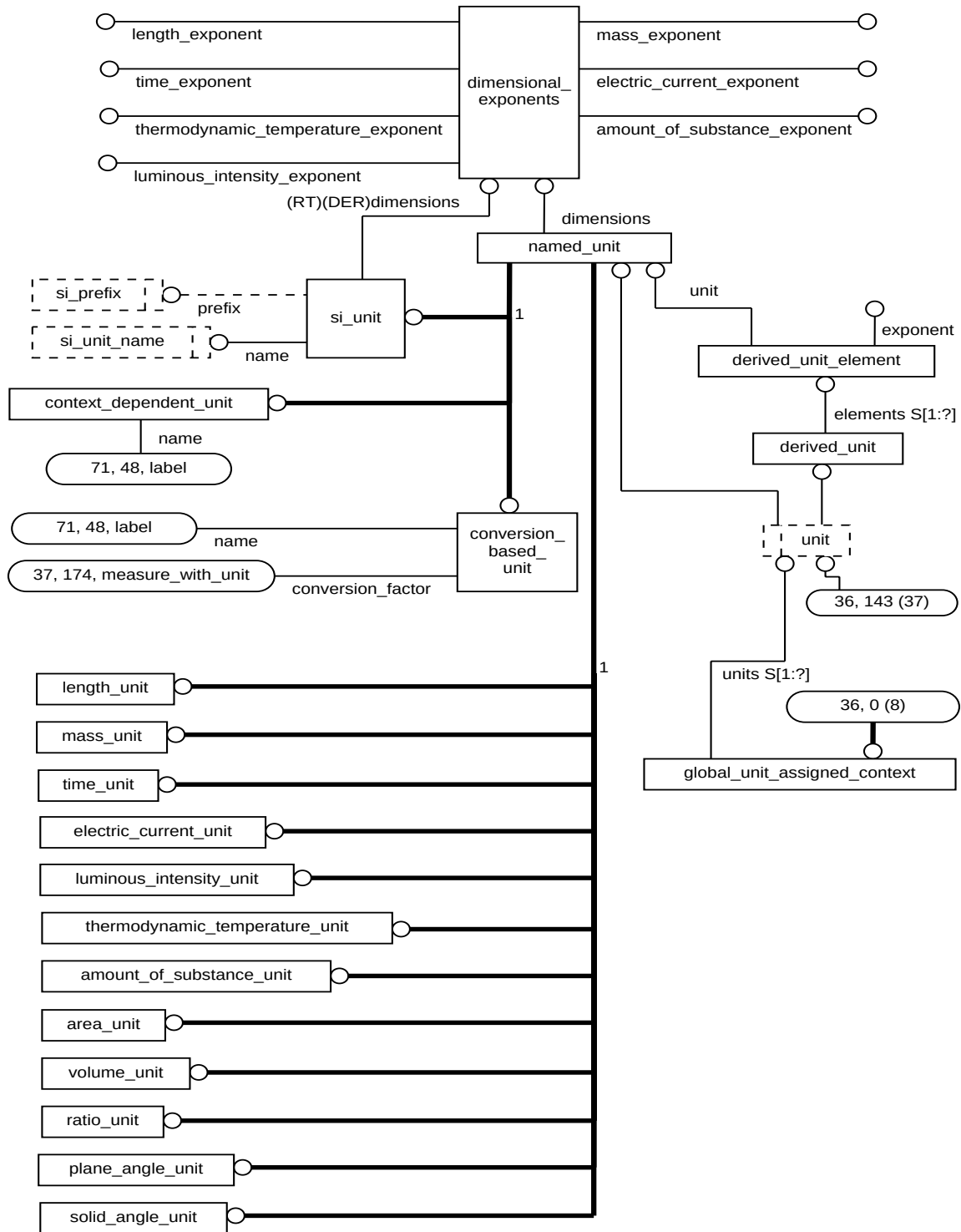


Figure J.36 – AIM EXPRESS-G diagram 36 of 80

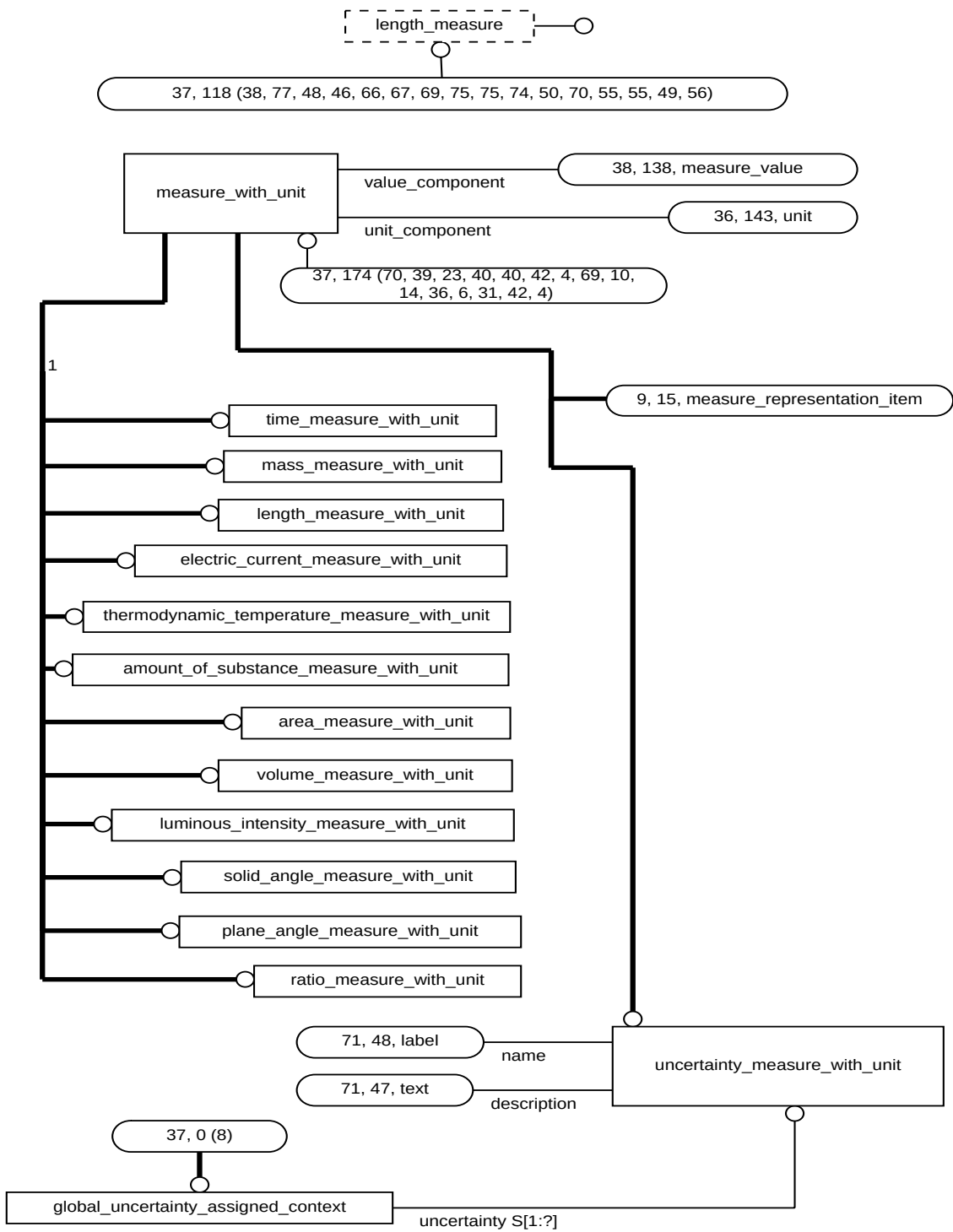


Figure J.37 – AIM EXPRESS-G diagram 37 of 80

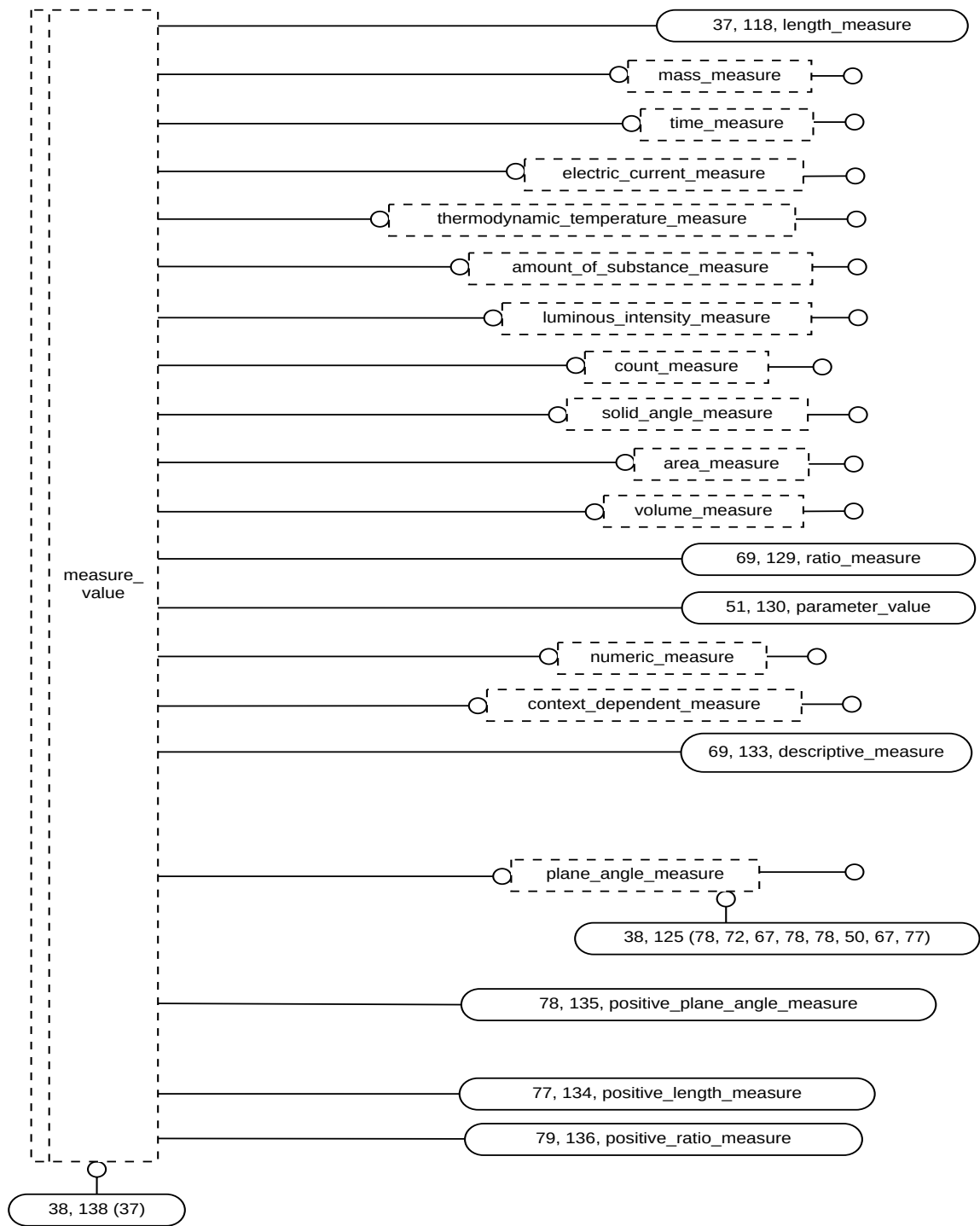


Figure J.38 – AIM EXPRESS-G diagram 38 of 80

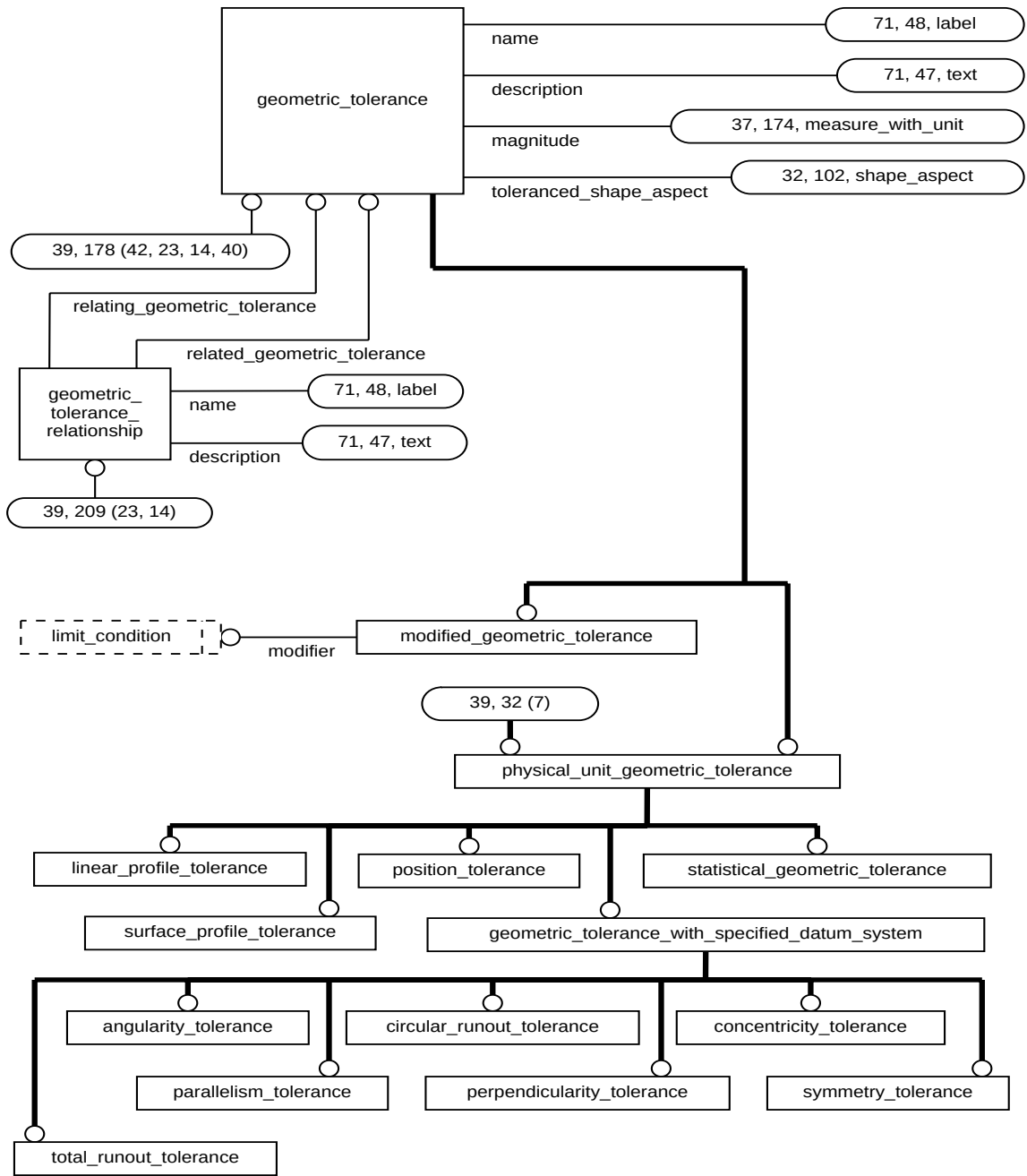


Figure J.39 – AIM EXPRESS-G diagram 39 of 80

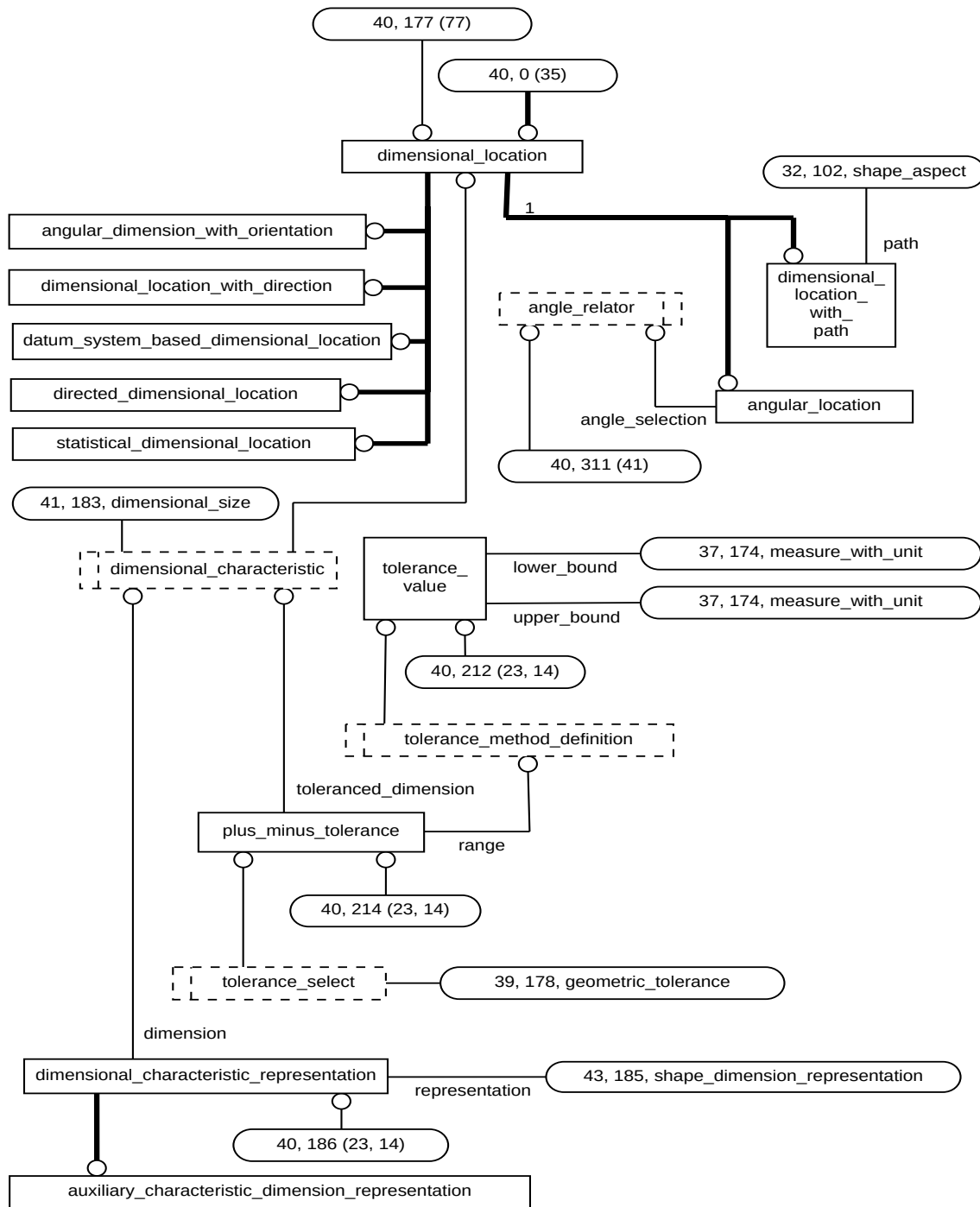


Figure J.40 – AIM EXPRESS-G diagram 40 of 80

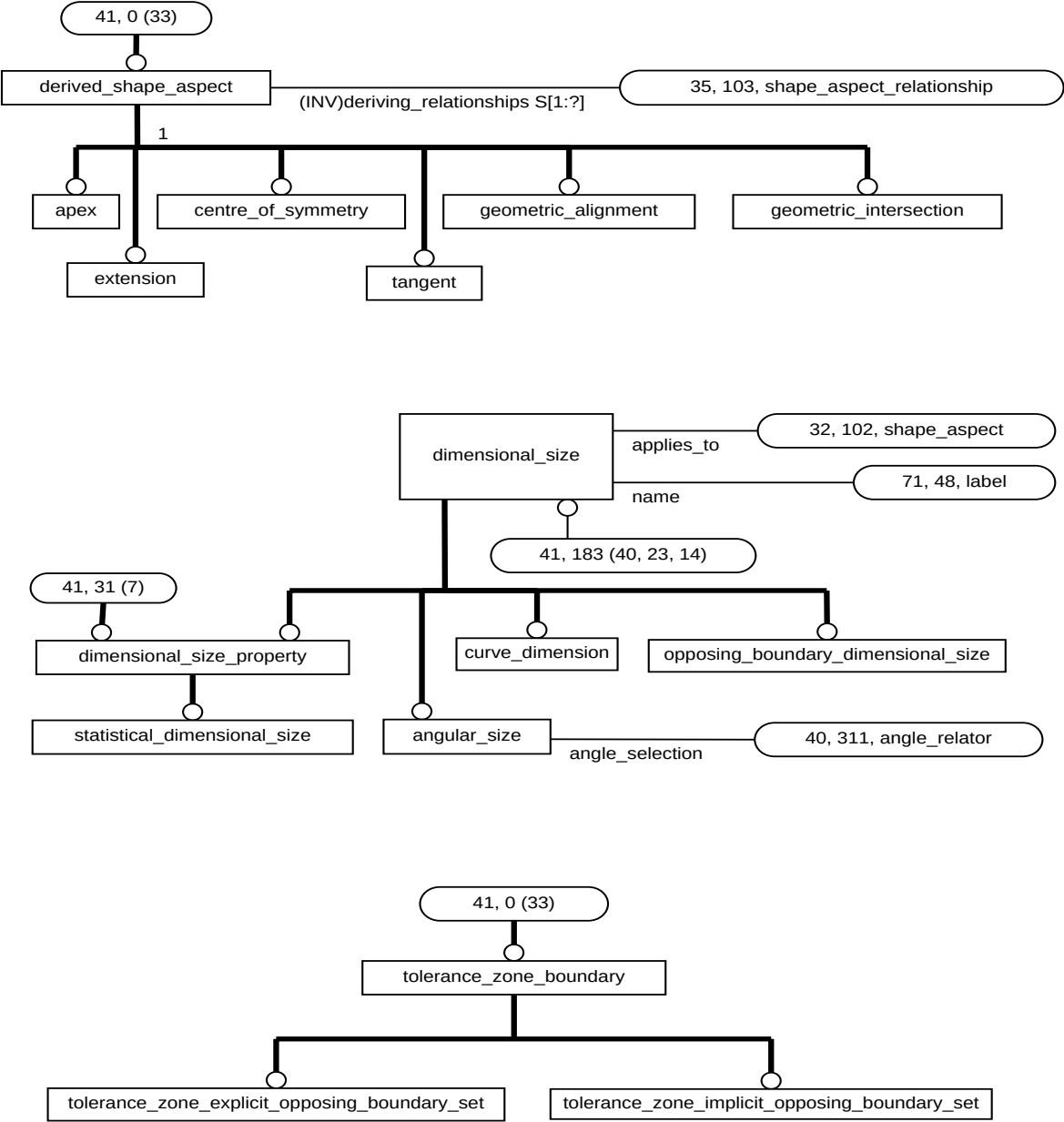


Figure J.41 – AIM EXPRESS-G diagram 41 of 80

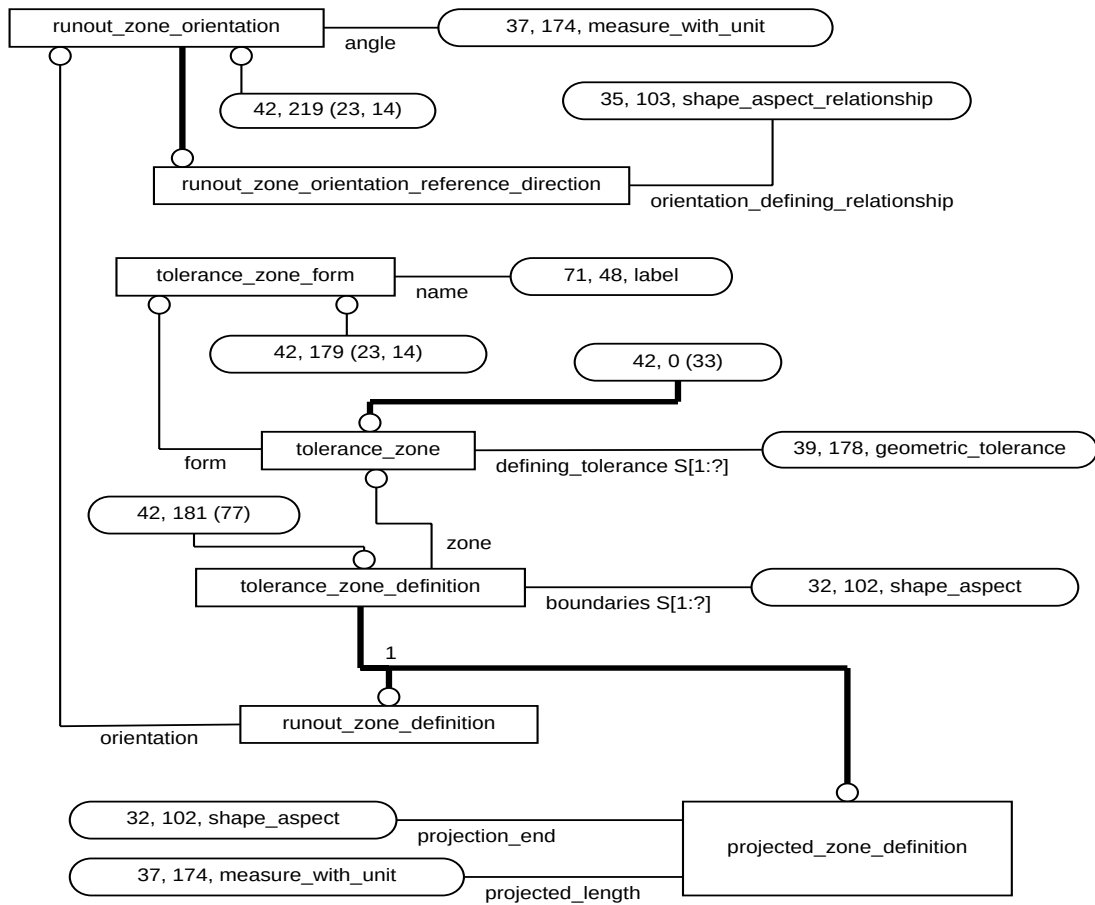


Figure J.42 – AIM EXPRESS-G diagram 42 of 80

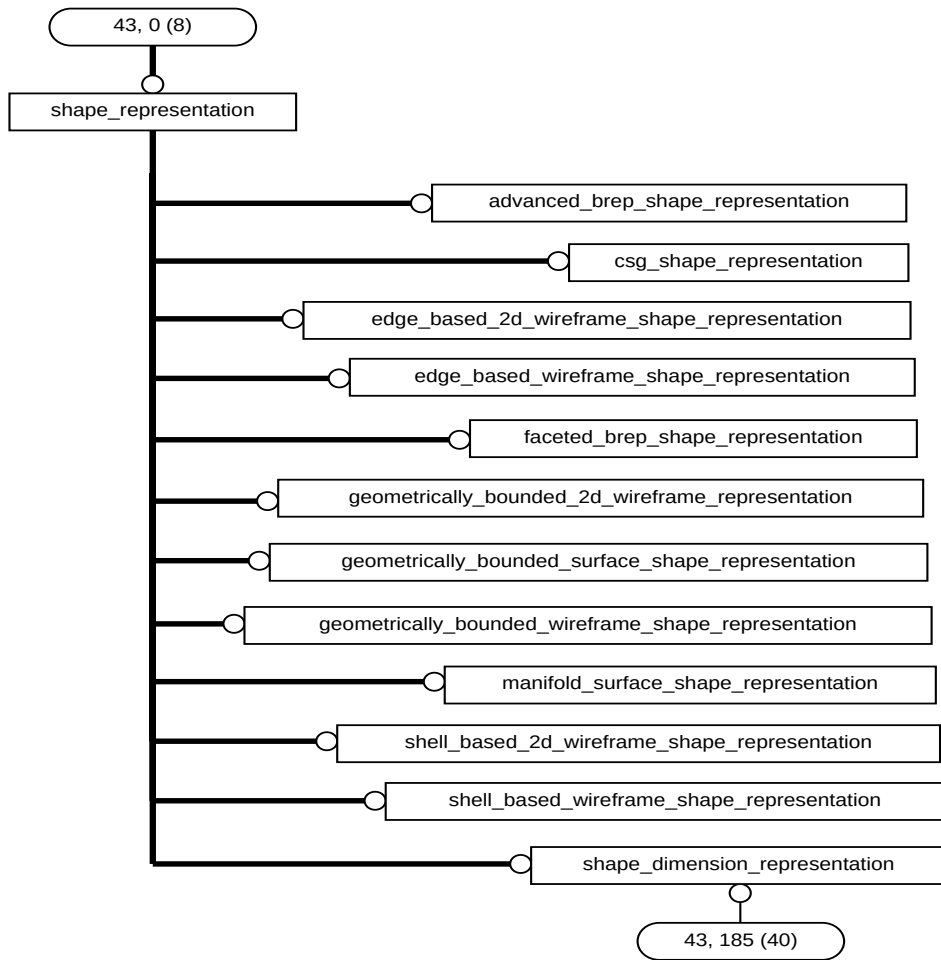


Figure J.43 – AIM EXPRESS-G diagram 43 of 80

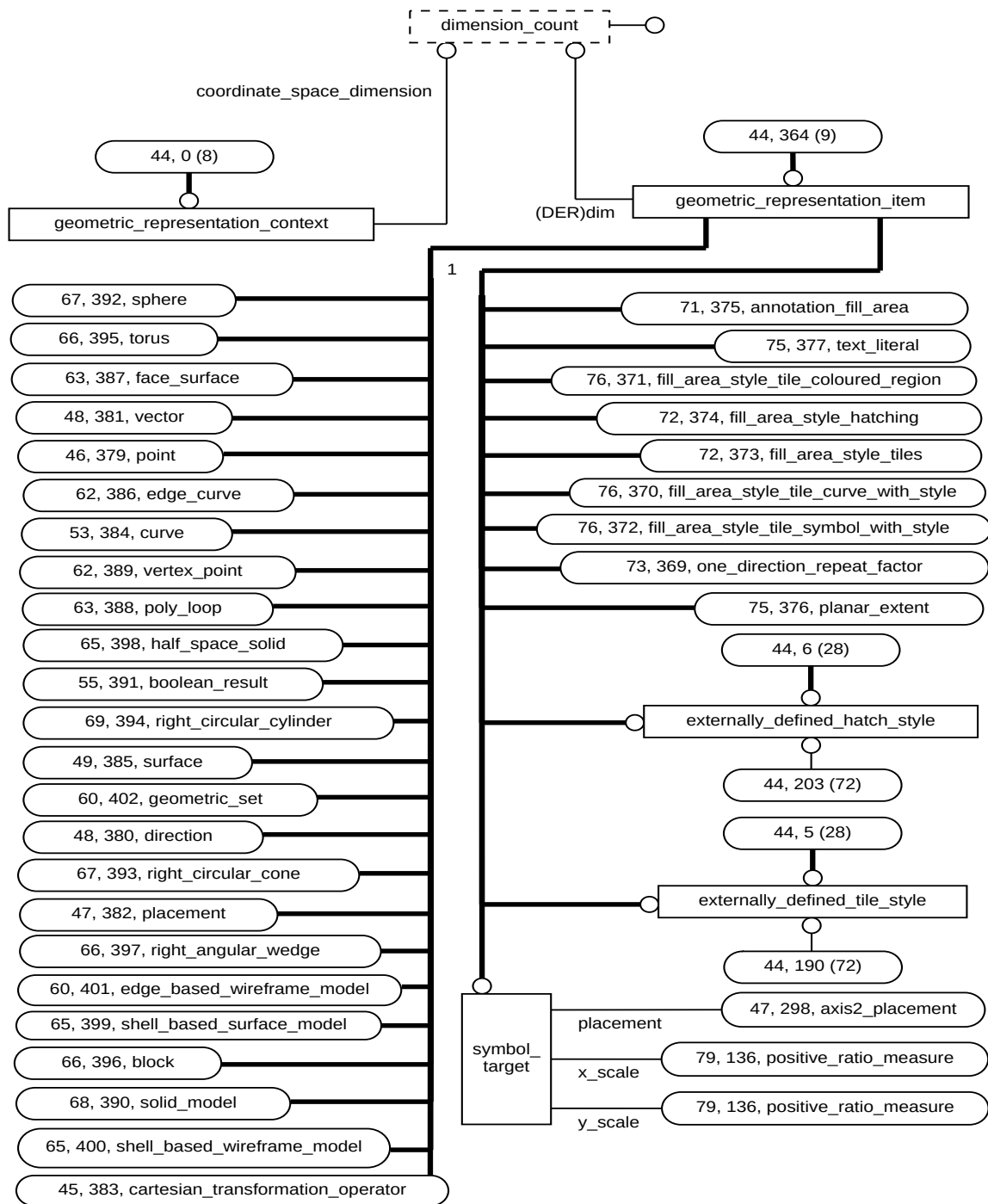


Figure J.44 – AIM EXPRESS-G diagram 44 of 80

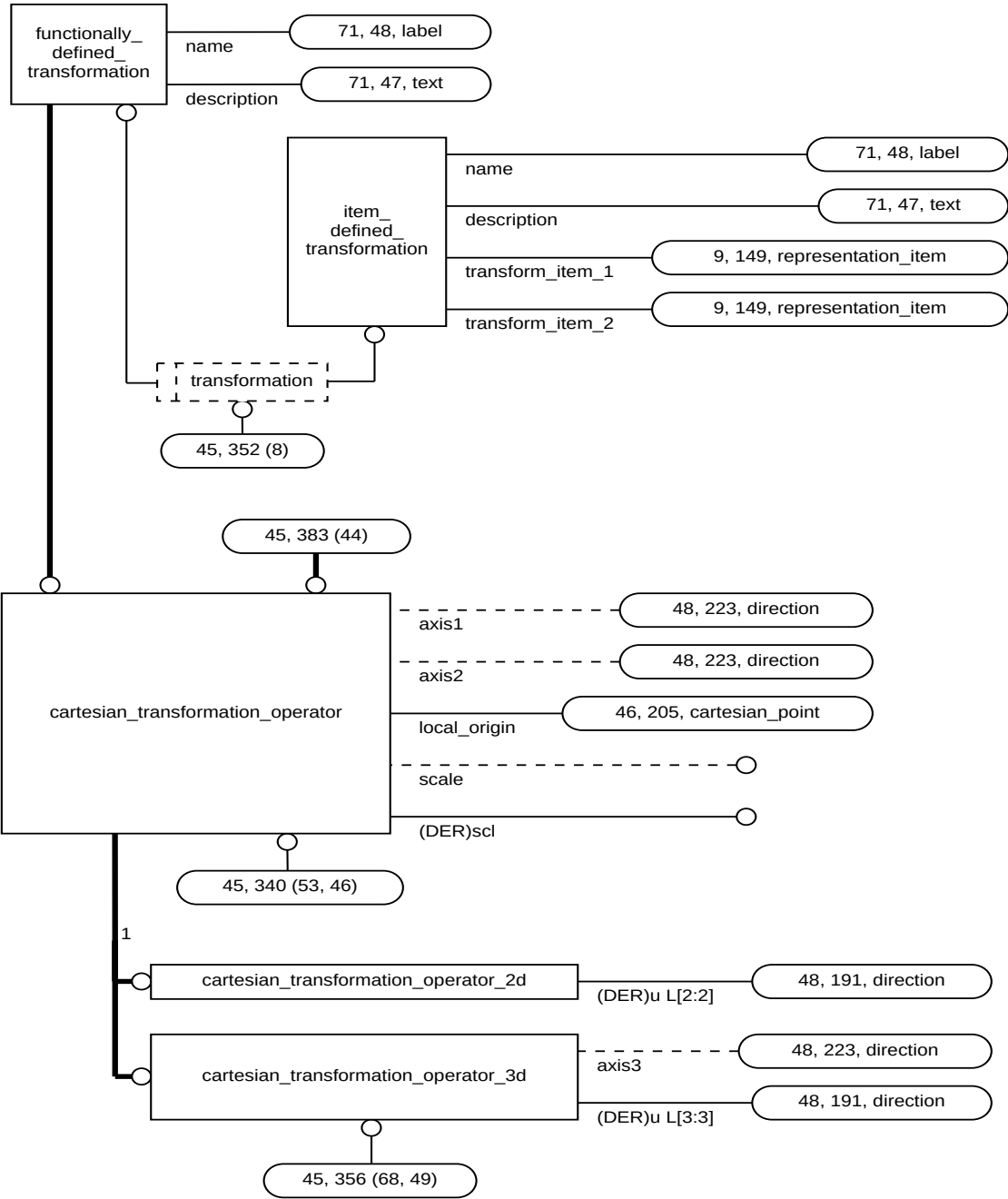


Figure J.45 – AIM EXPRESS-G diagram 45 of 80

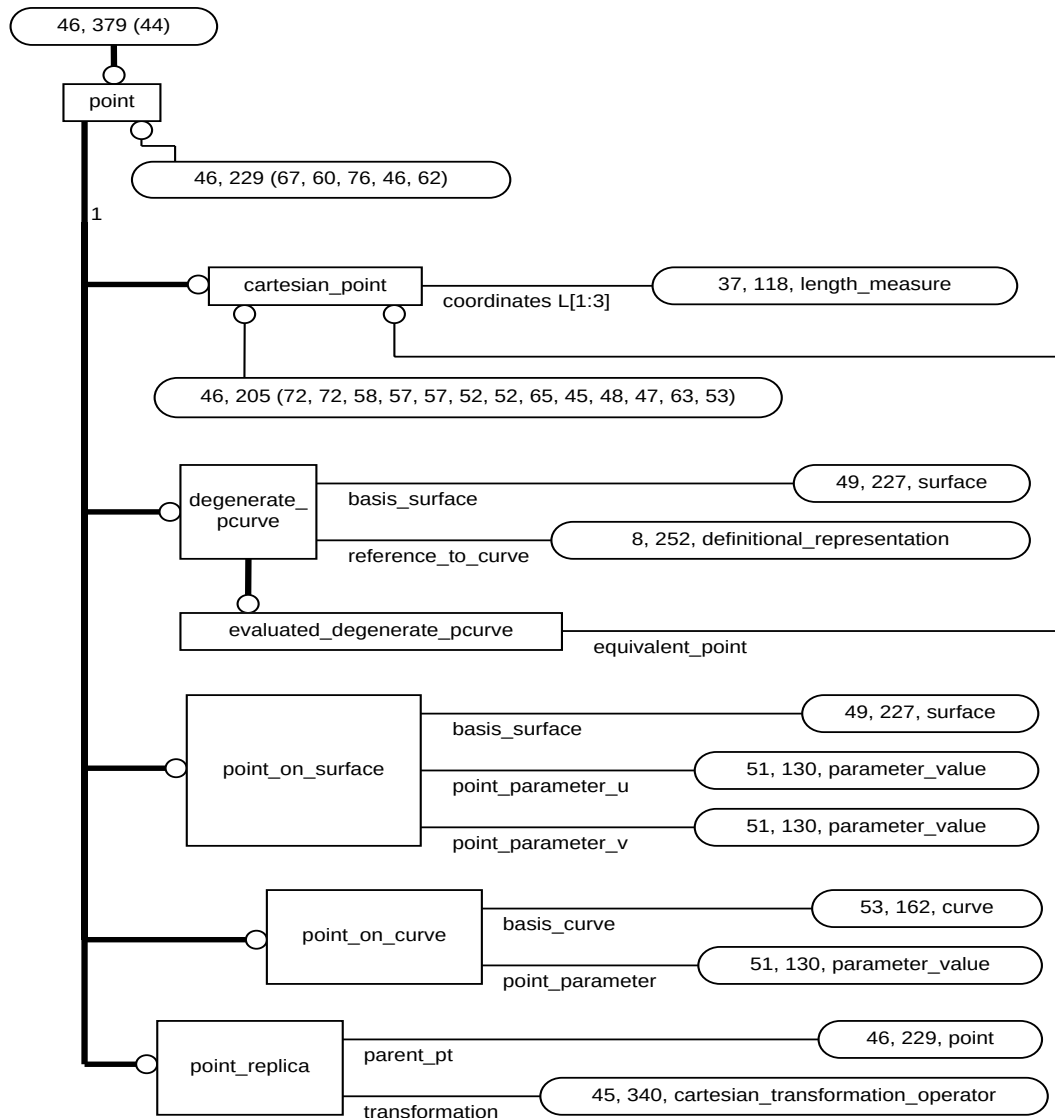


Figure J.46 – AIM EXPRESS-G diagram 46 of 80

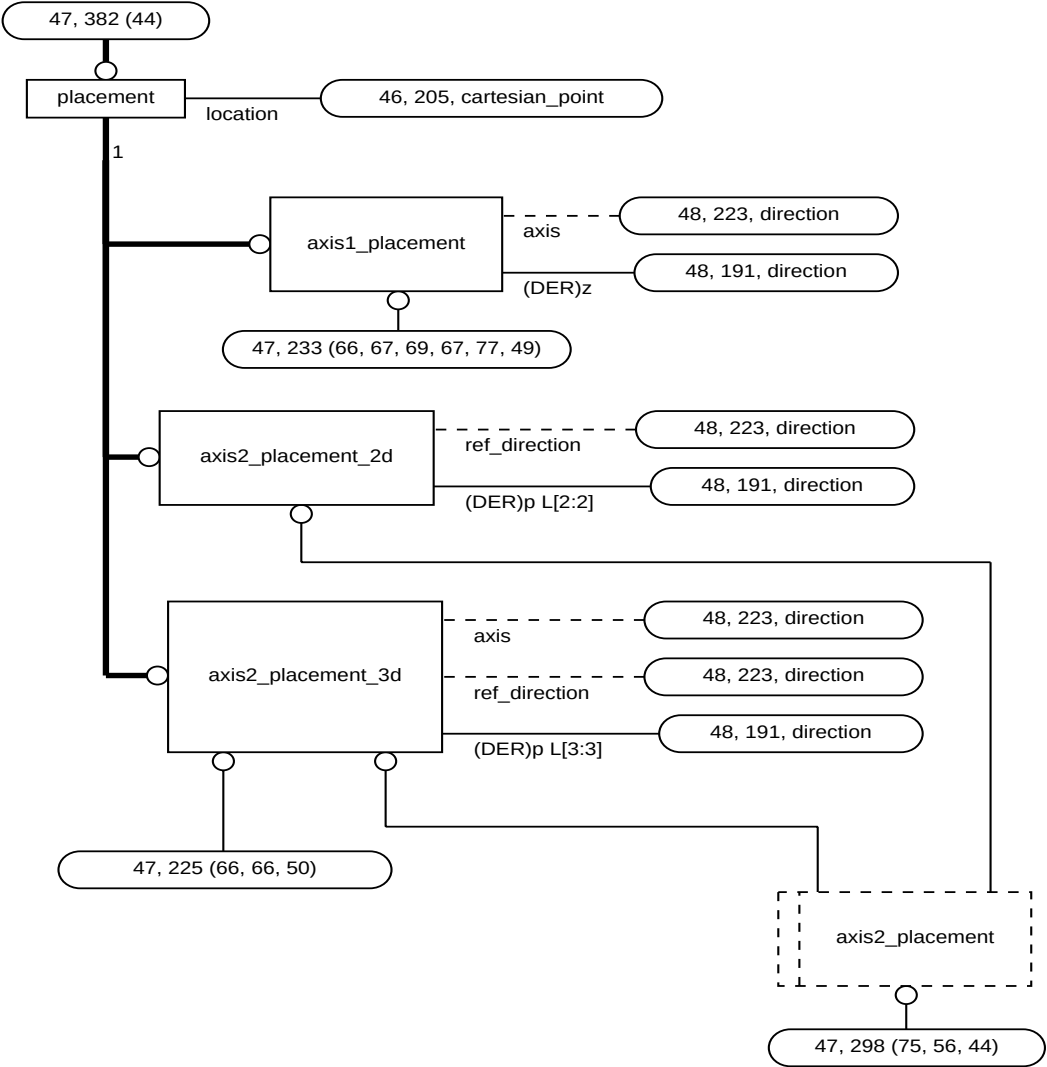


Figure J.47 – AIM EXPRESS-G diagram 47 of 80

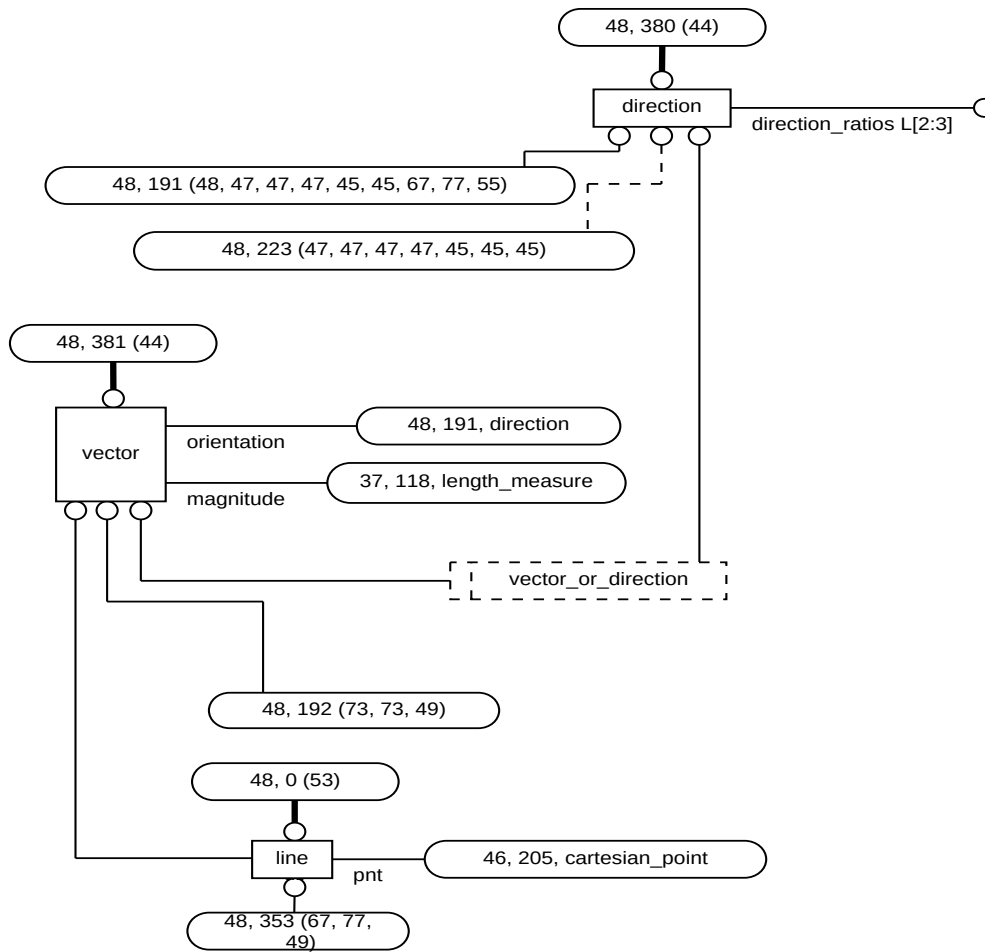


Figure J.48 – AIM EXPRESS-G diagram 48 of 80

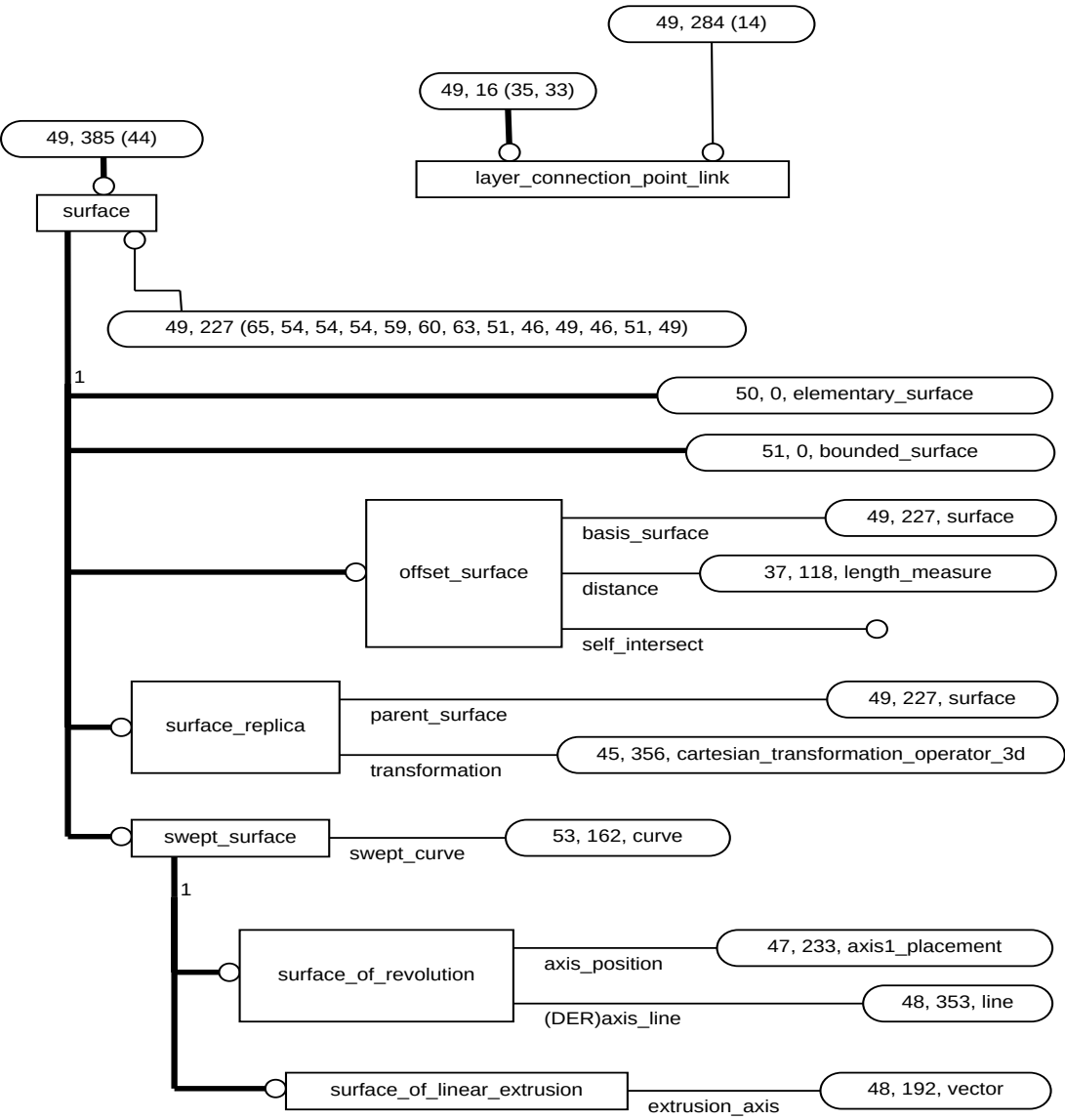


Figure J.49 – AIM EXPRESS-G diagram 49 of 80

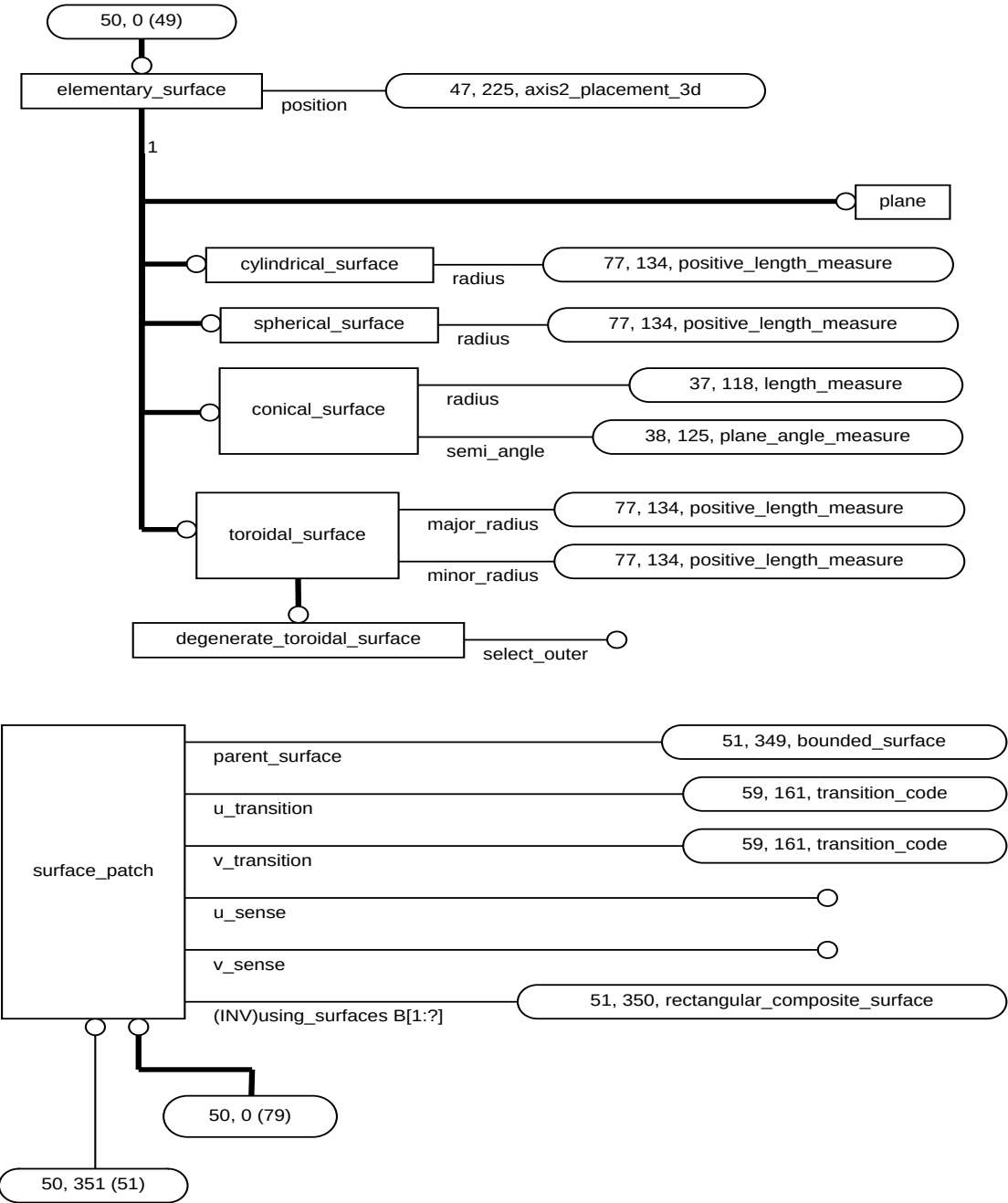


Figure J.50 – AIM EXPRESS-G diagram 50 of 80

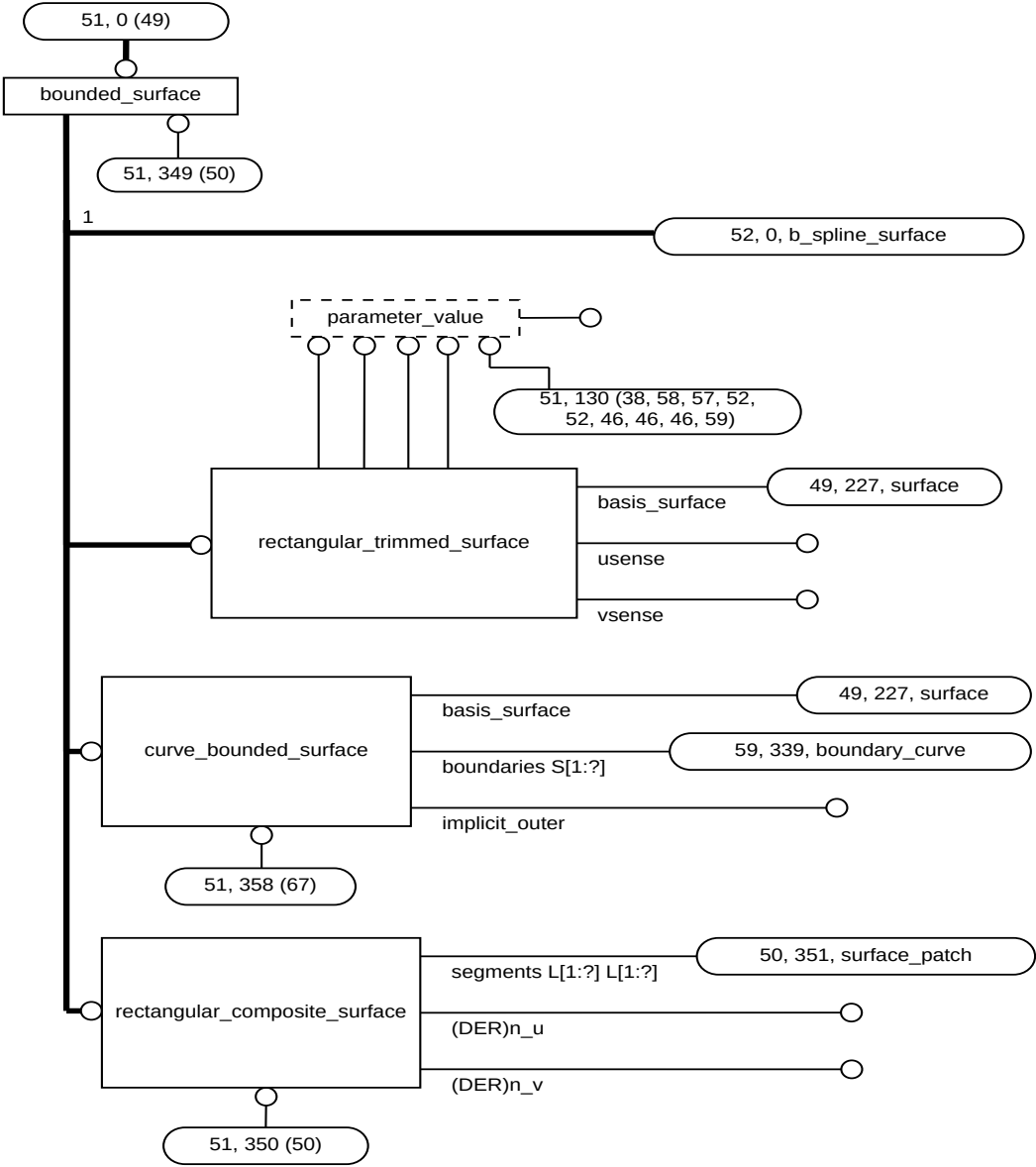


Figure J.51 – AIM EXPRESS-G diagram 51 of 80

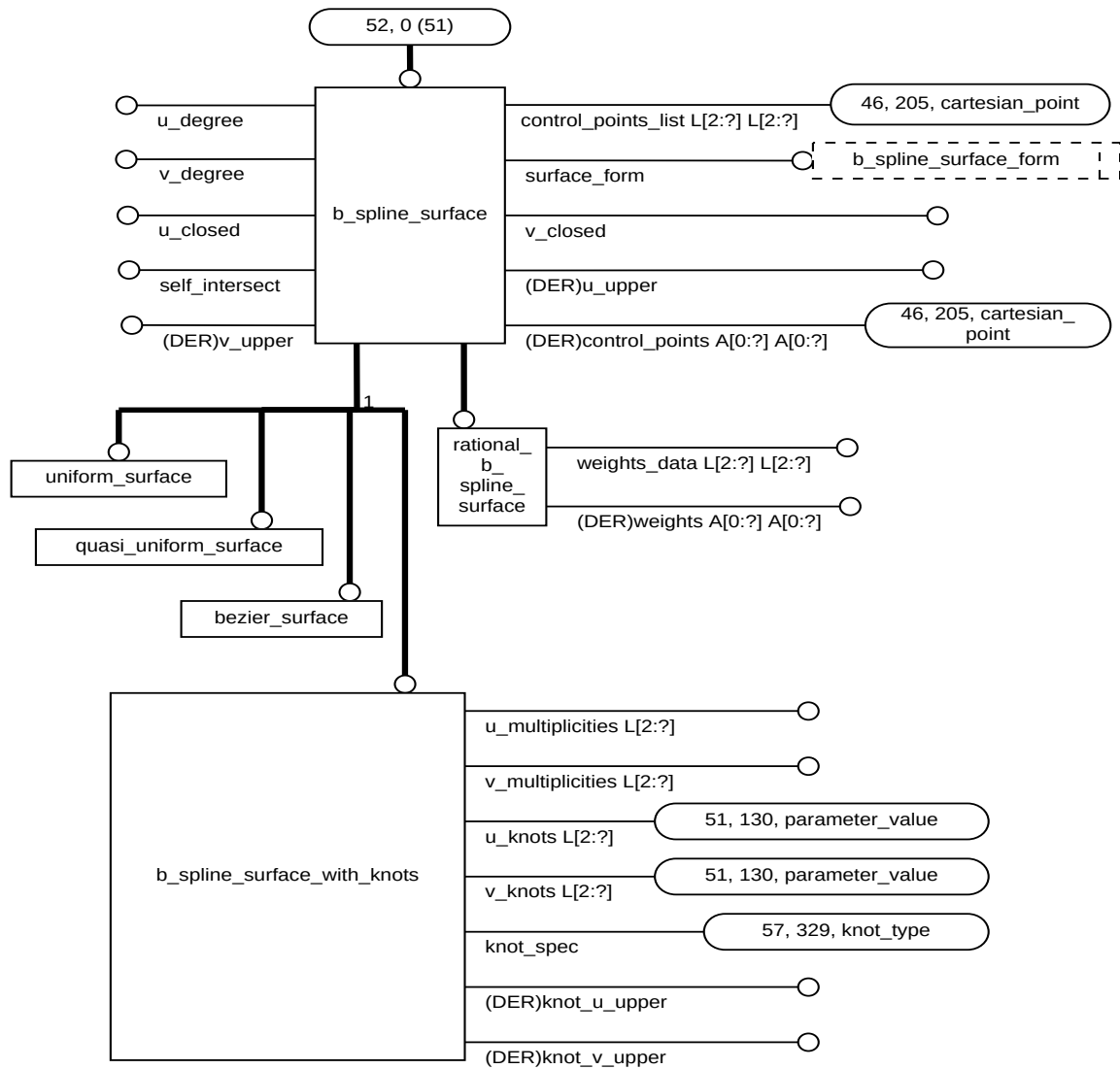


Figure J.52 – AIM EXPRESS-G diagram 52 of 80

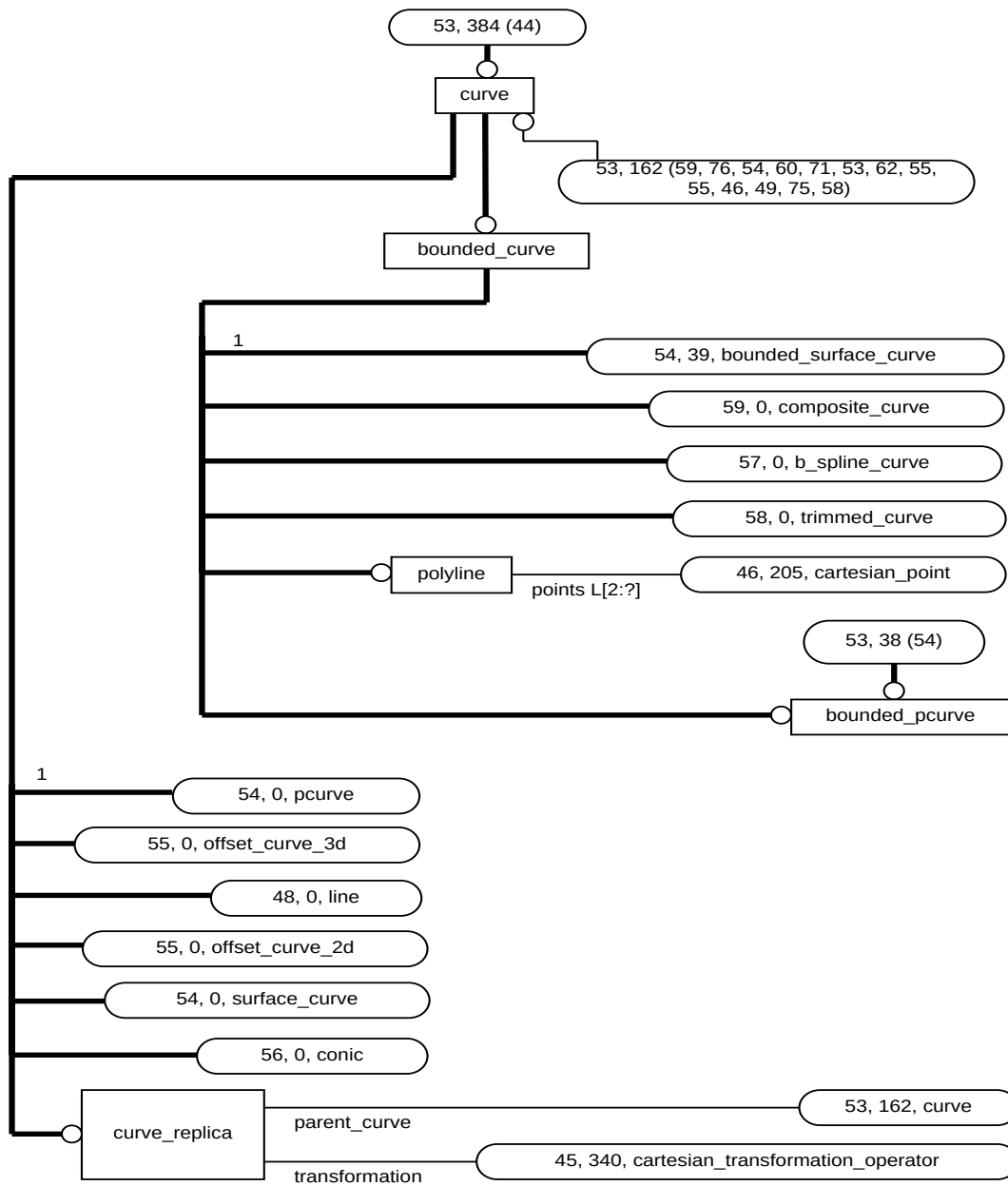


Figure J.53 – AIM EXPRESS-G diagram 53 of 80

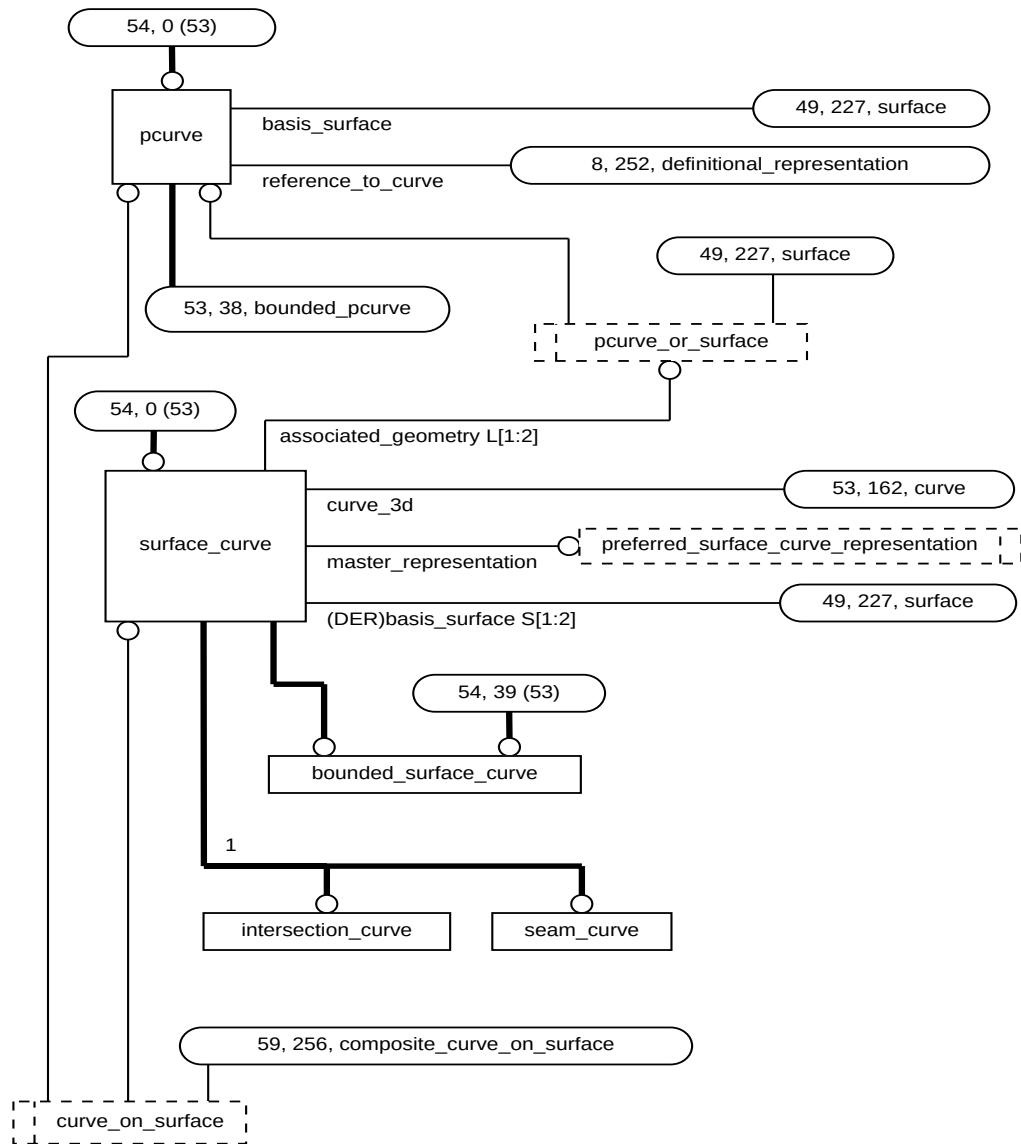


Figure J.54 – AIM EXPRESS-G diagram 54 of 80

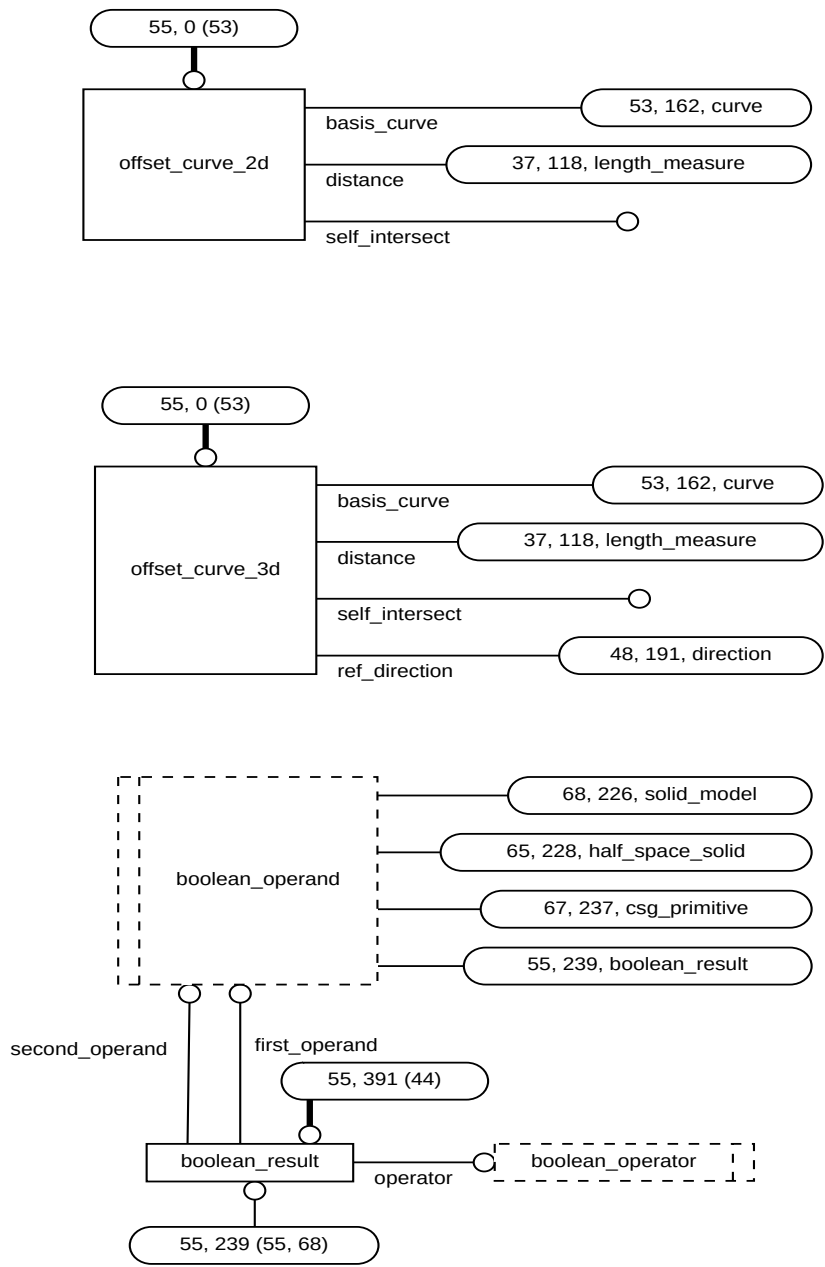


Figure J.55 – AIM EXPRESS-G diagram 55 of 80

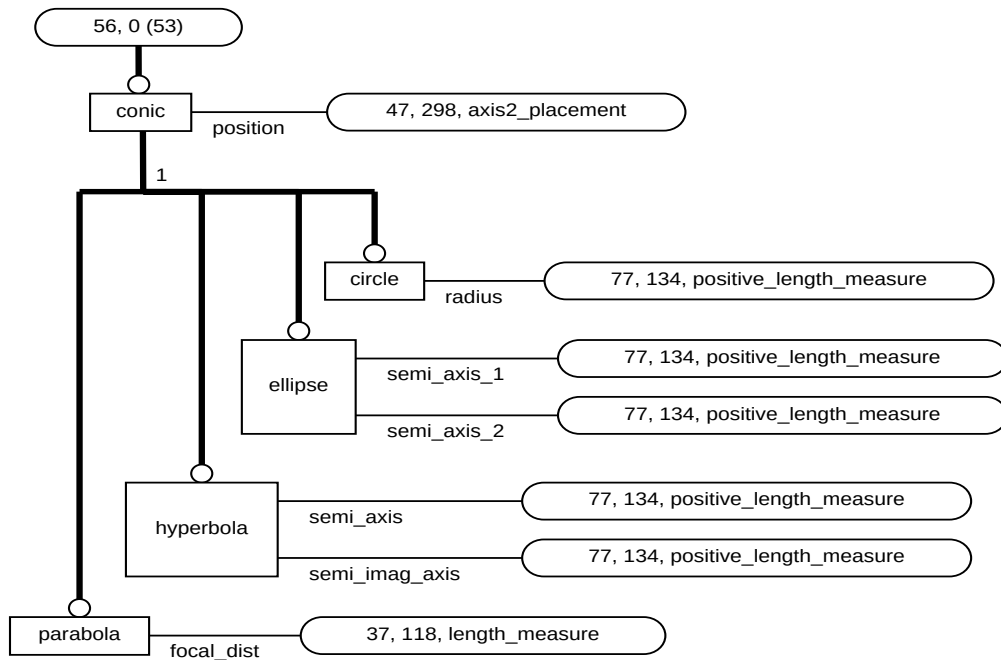


Figure J.56 – AIM EXPRESS-G diagram 56 of 80

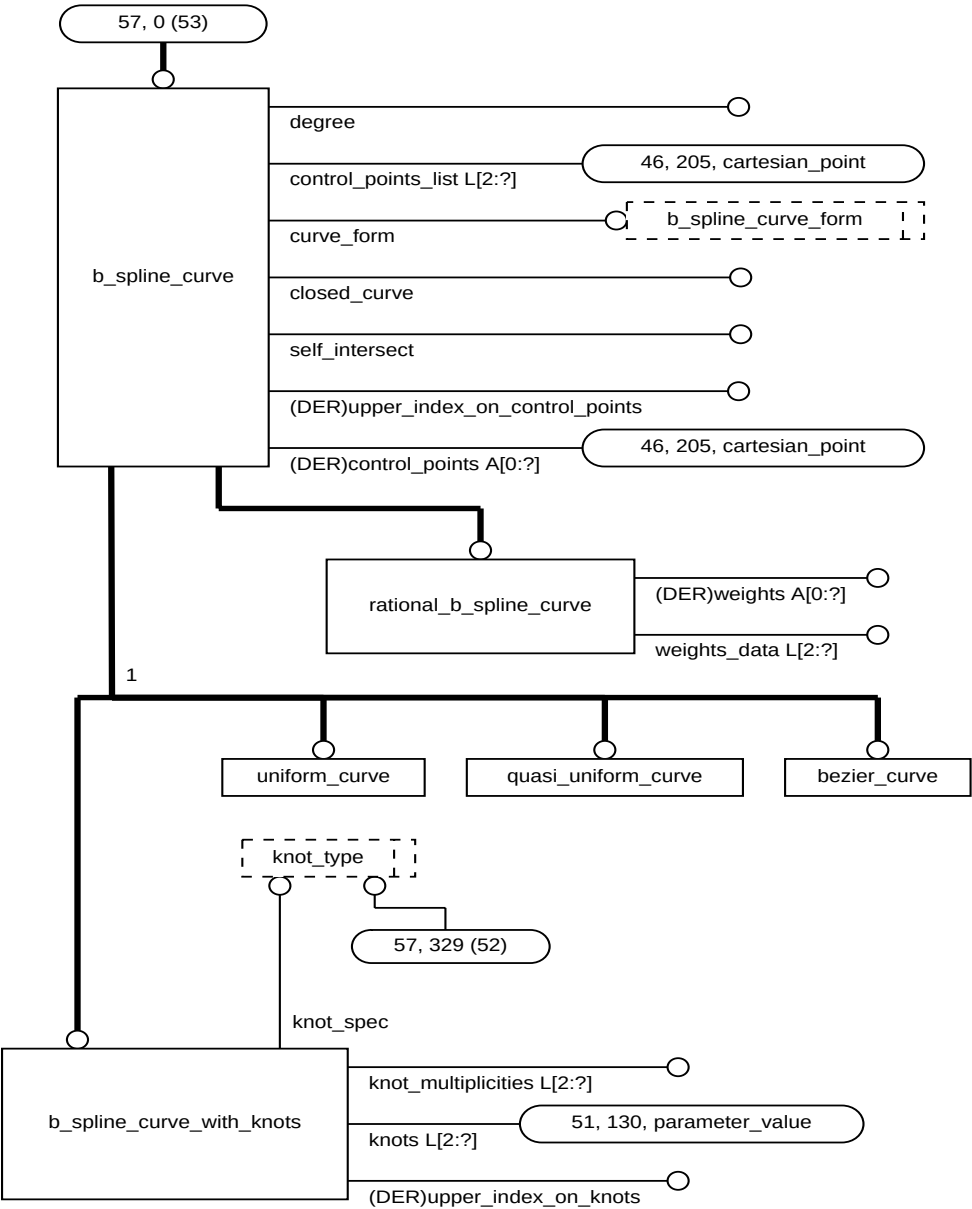


Figure J.57 – AIM EXPRESS-G diagram 57 of 80

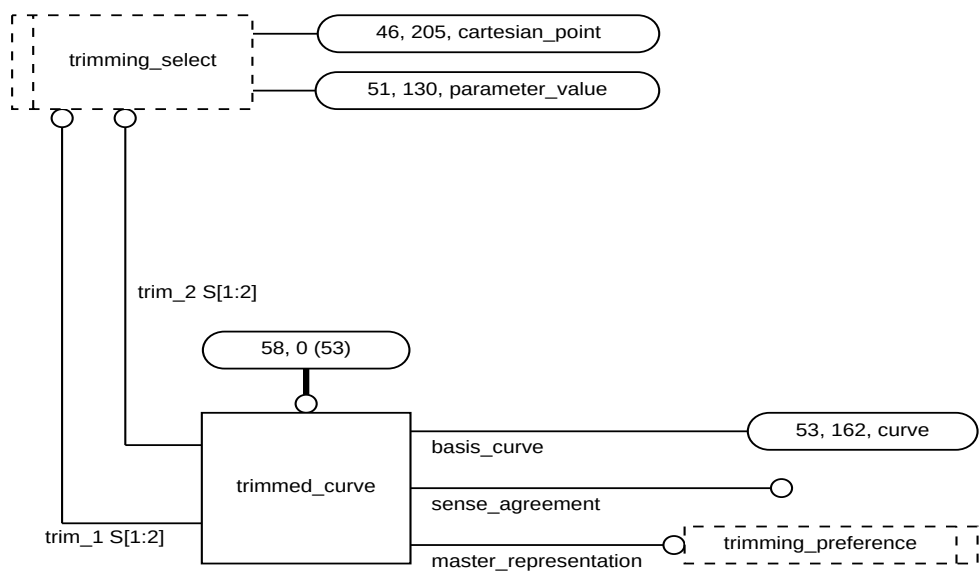


Figure J.58 – AIM EXPRESS-G diagram 58 of 80

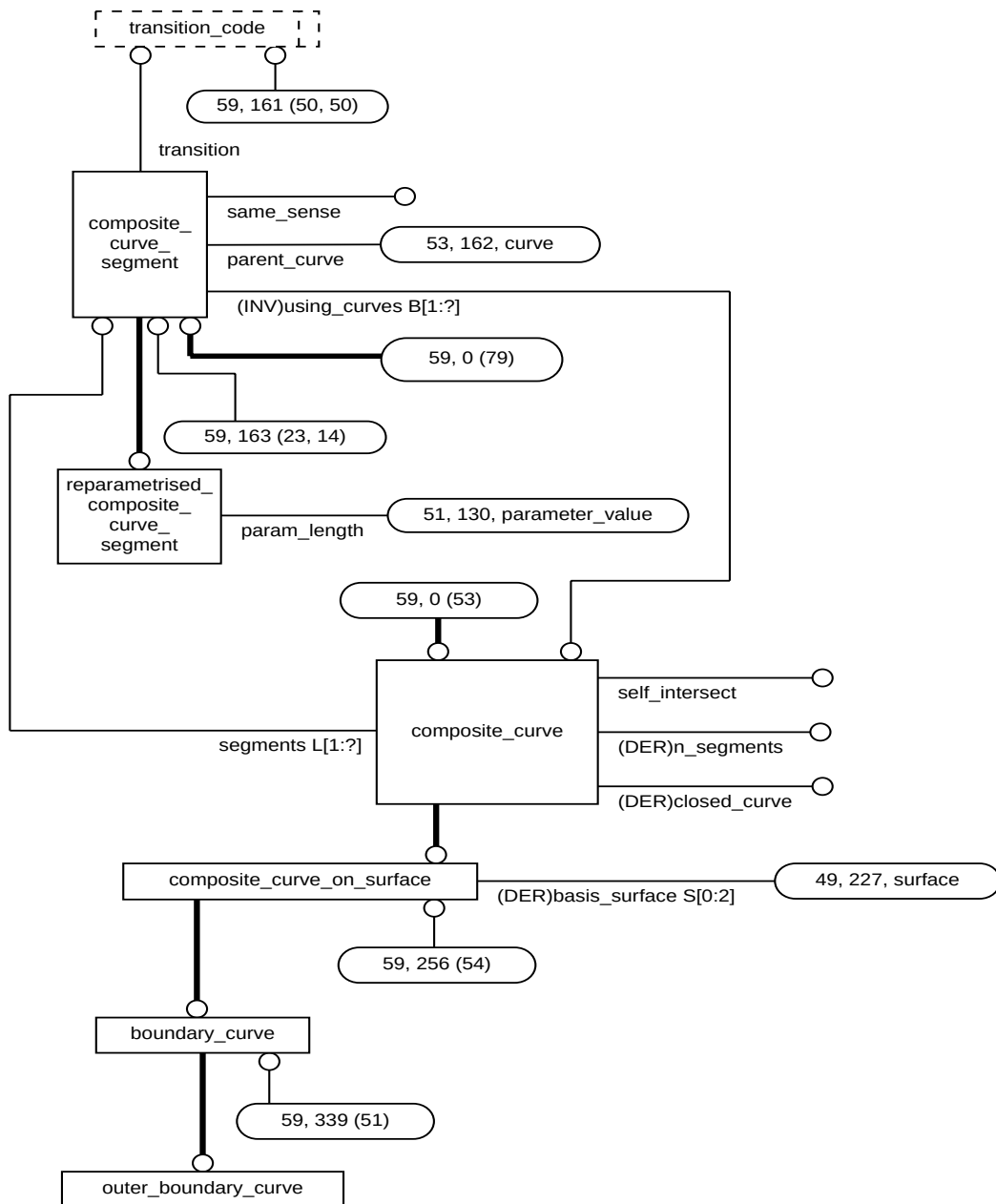


Figure J.59 – AIM EXPRESS-G diagram 59 of 80

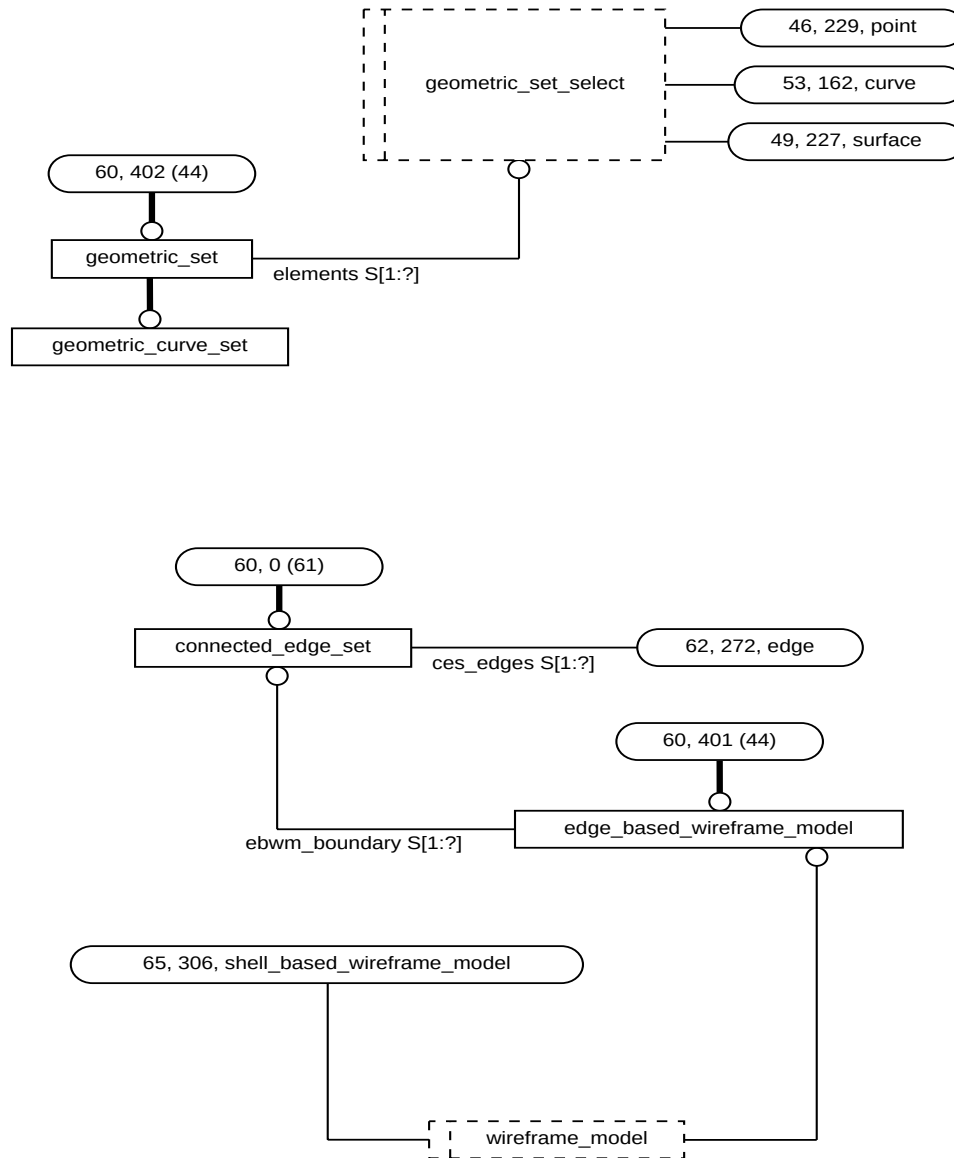


Figure J.60 – AIM EXPRESS-G diagram 60 of 80

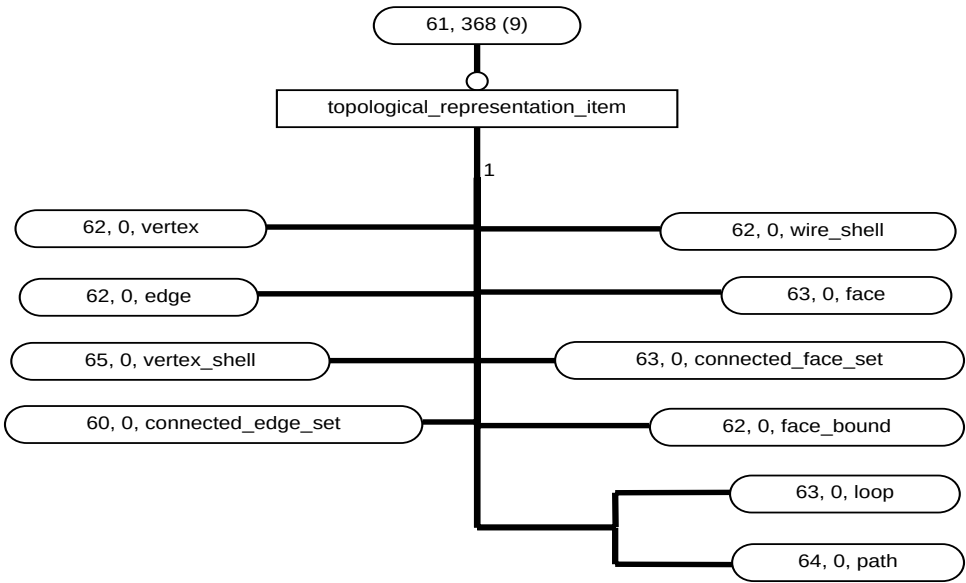


Figure J.61 – AIM EXPRESS-G diagram 61 of 80

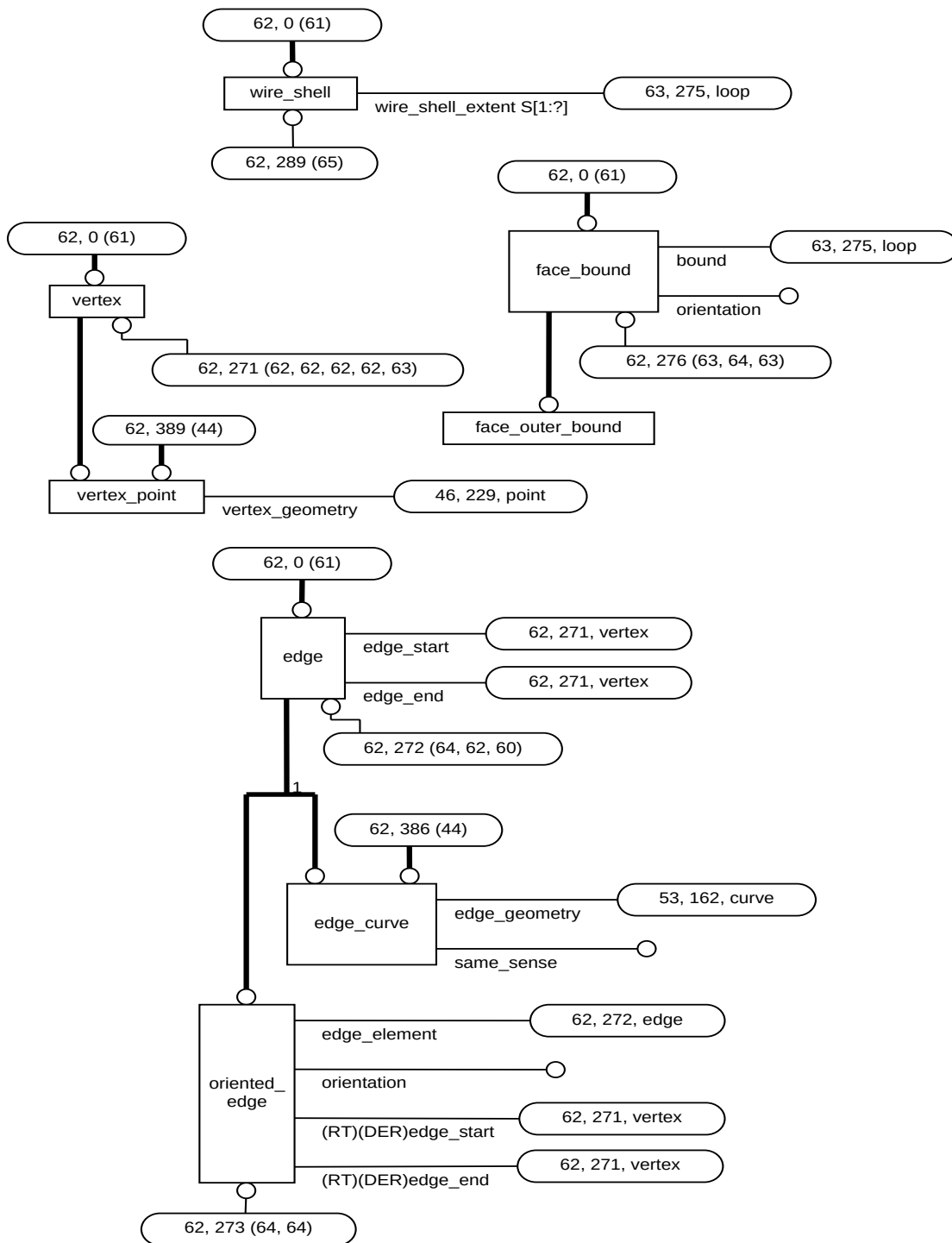


Figure J.62 – AIM EXPRESS-G diagram 62 of 80

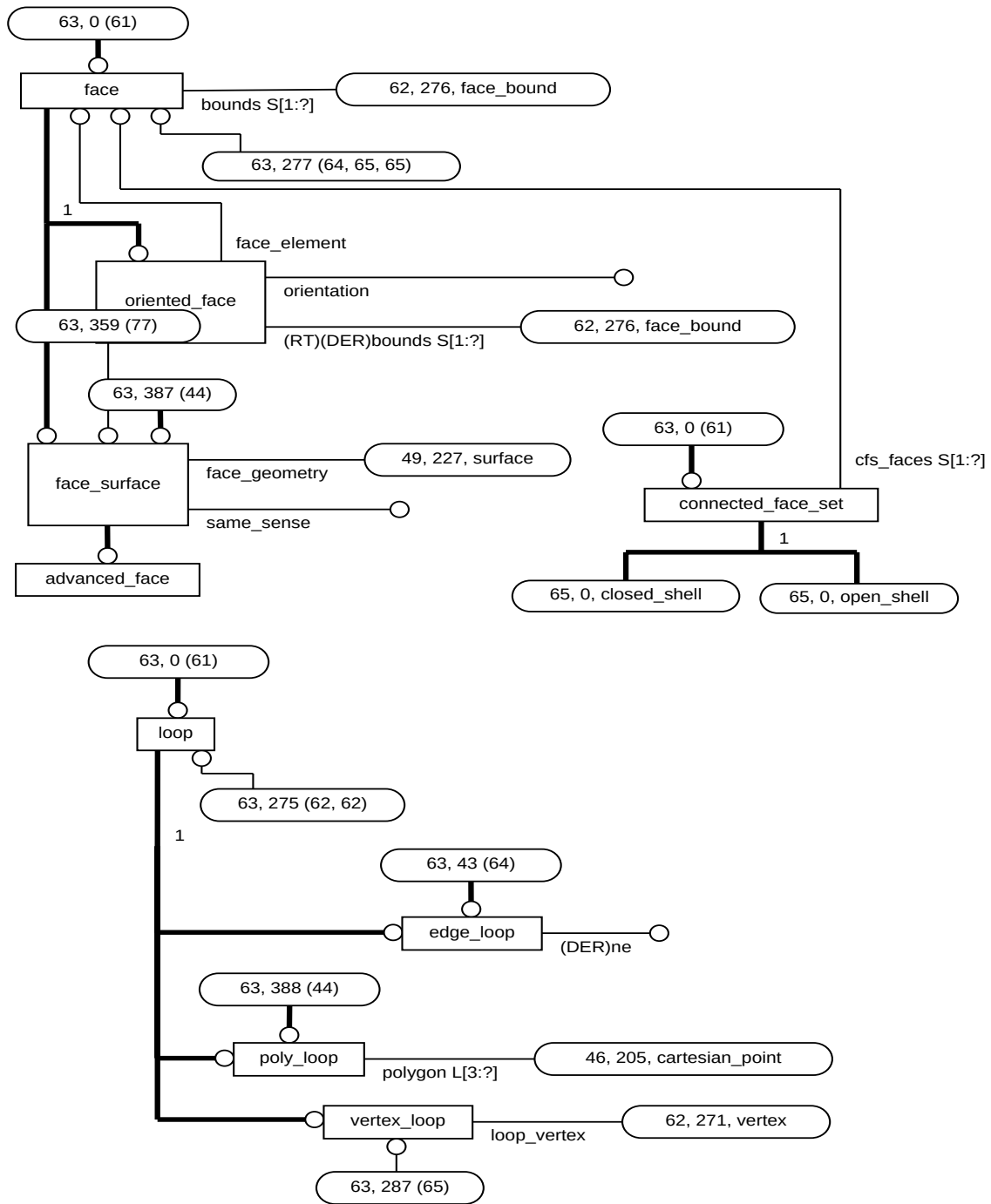


Figure J.63 – AIM EXPRESS-G diagram 63 of 80

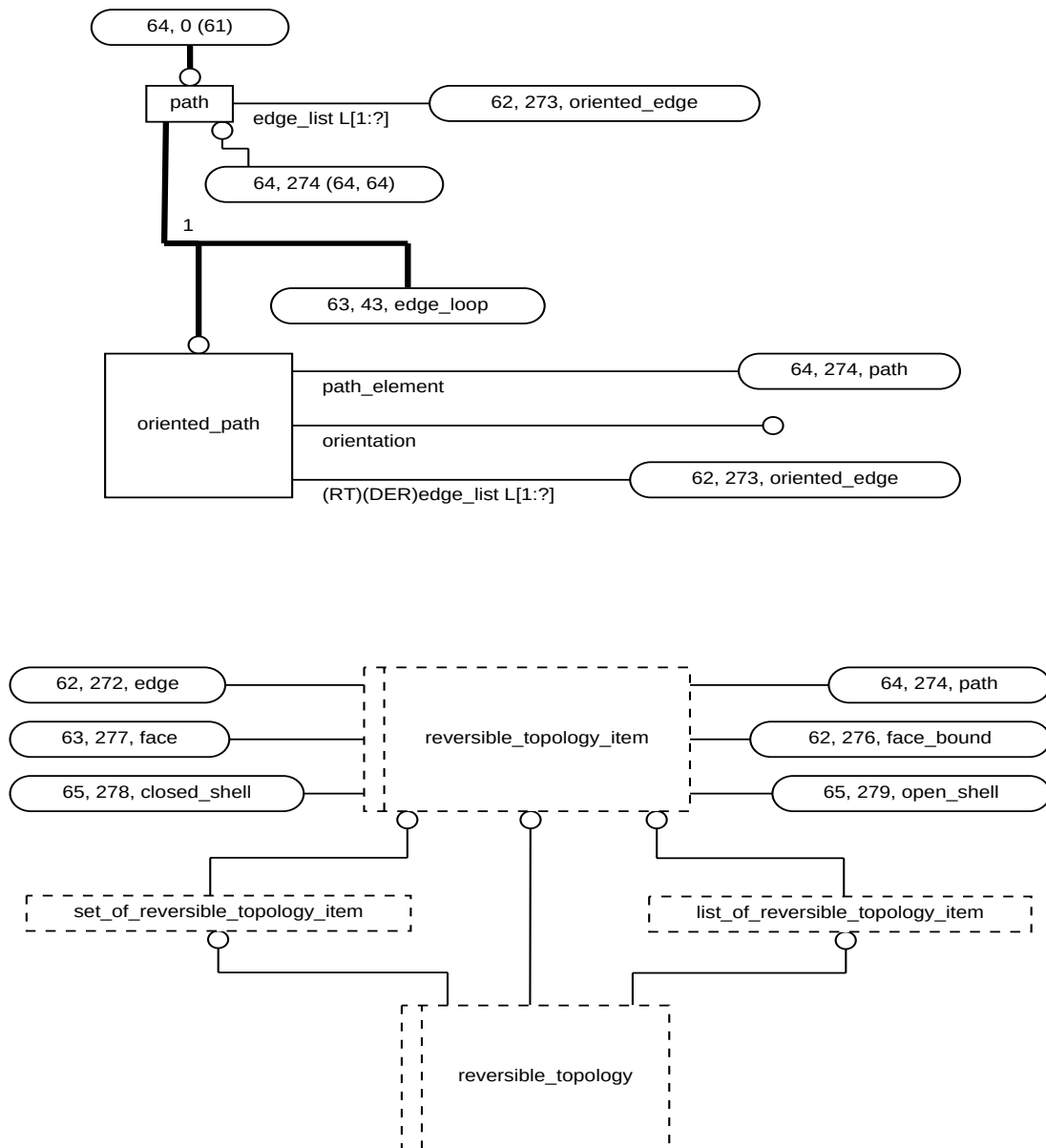


Figure J.64 – AIM EXPRESS-G diagram 64 of 80

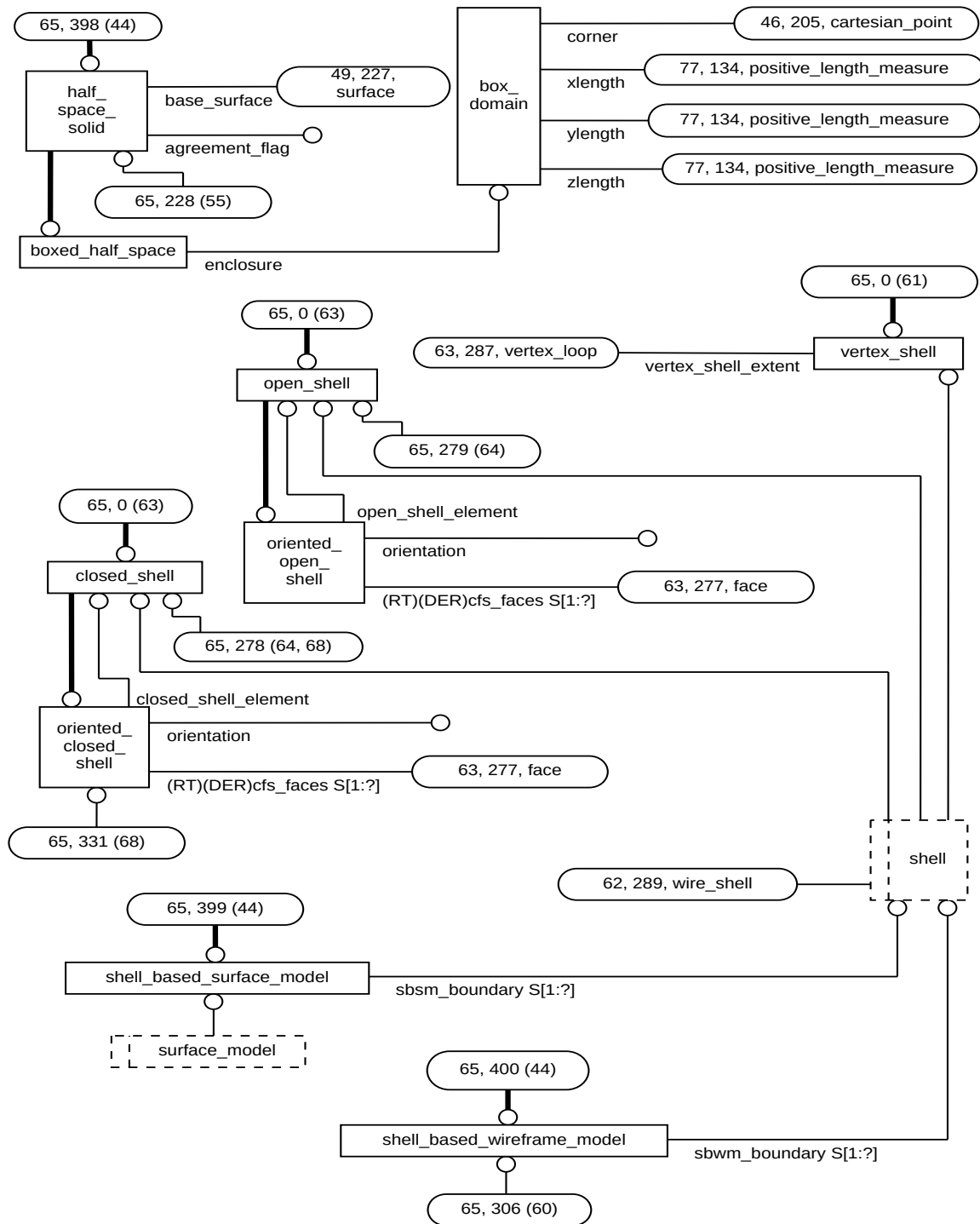


Figure J.65 – AIM EXPRESS-G diagram 65 of 80

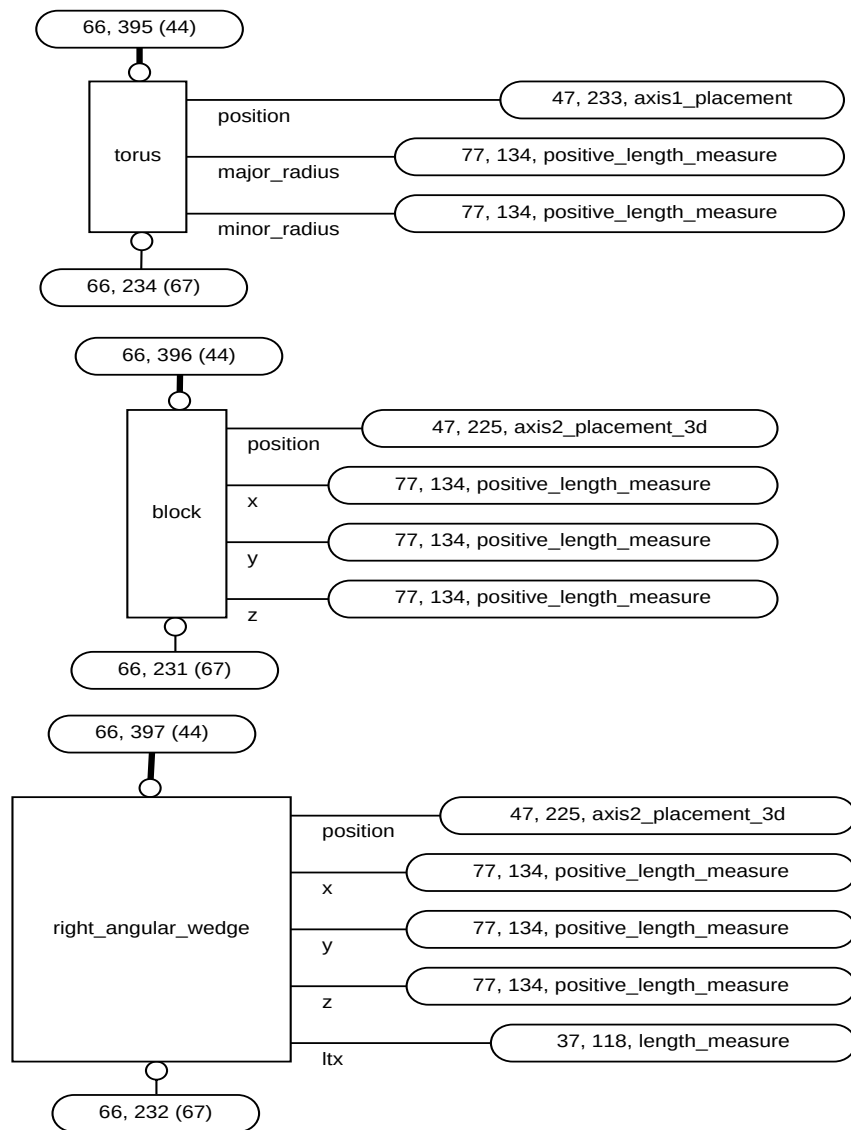


Figure J.66 – AIM EXPRESS-G diagram 66 of 80

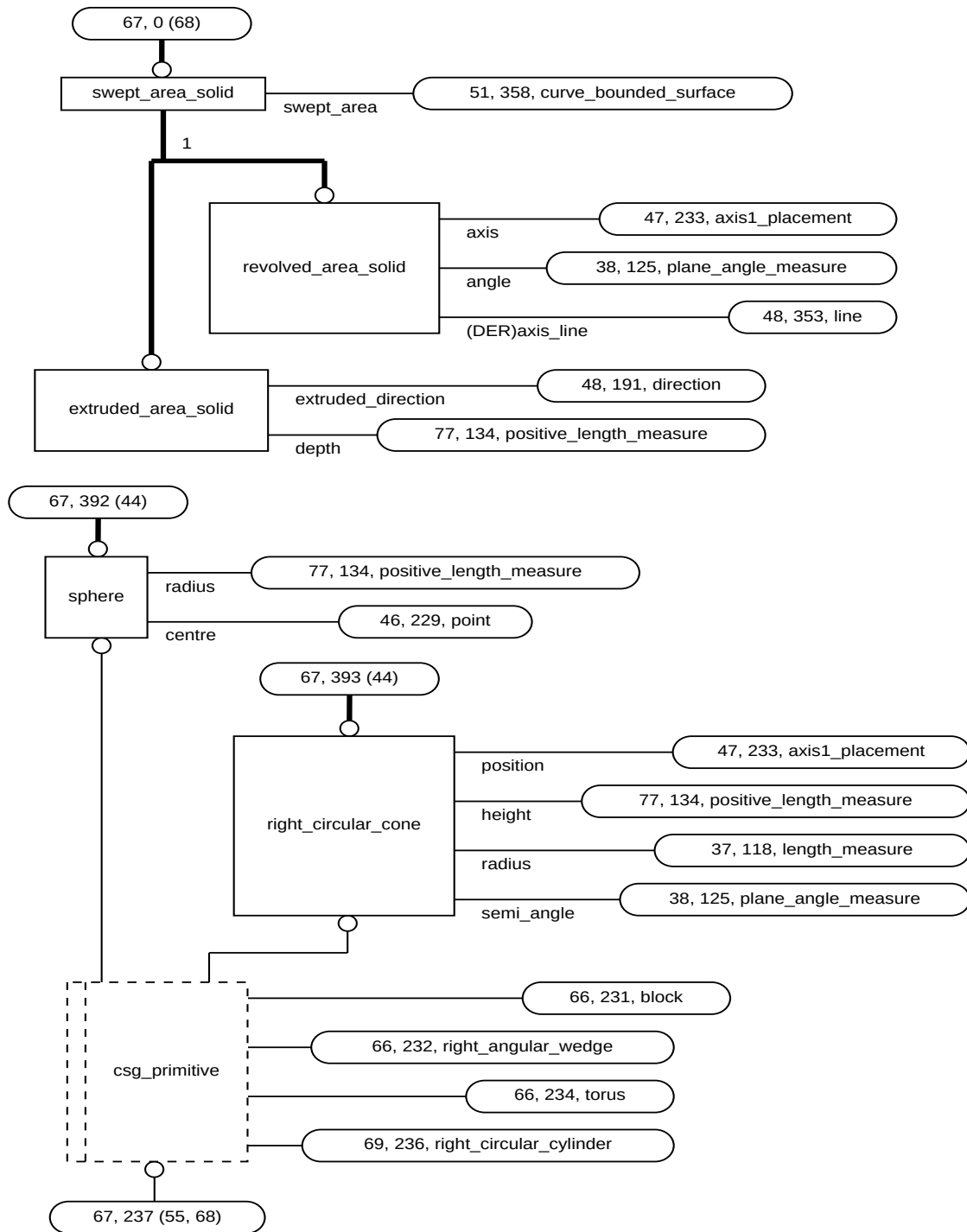


Figure J.67 – AIM EXPRESS-G diagram 67 of 80

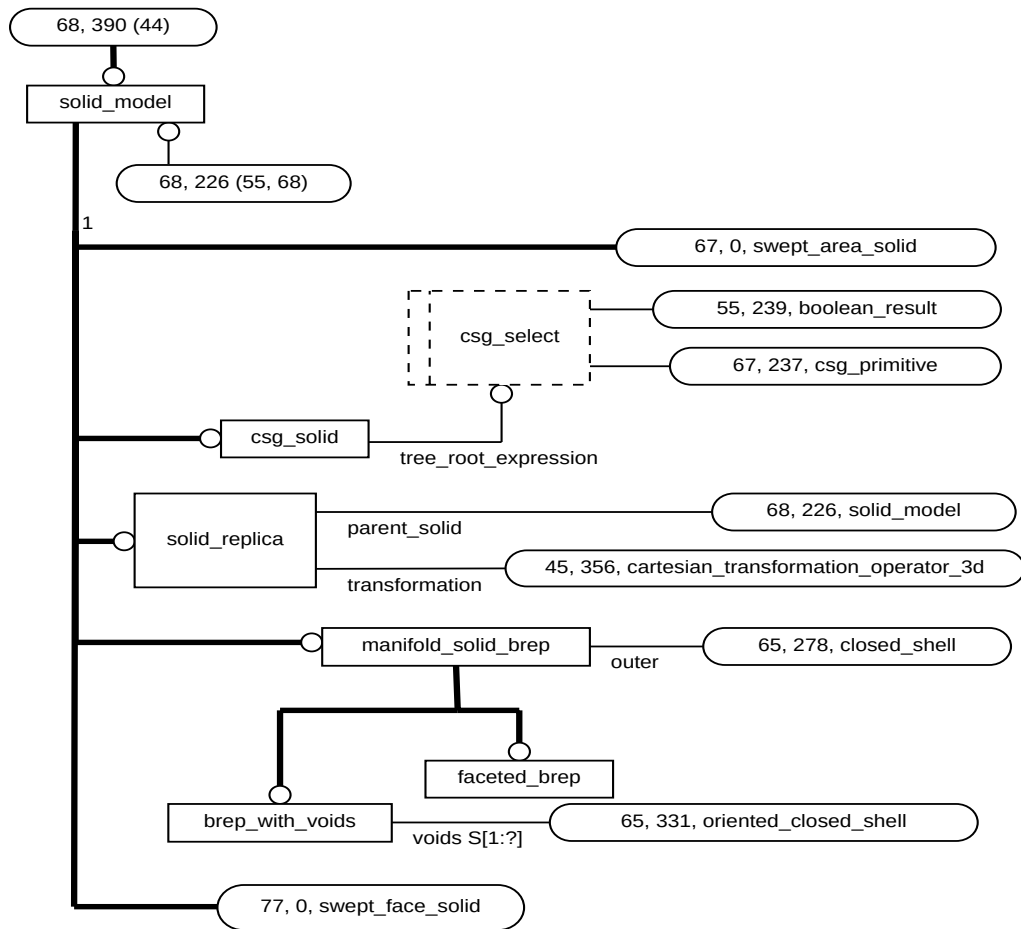


Figure J.68 – AIM EXPRESS-G diagram 68 of 80

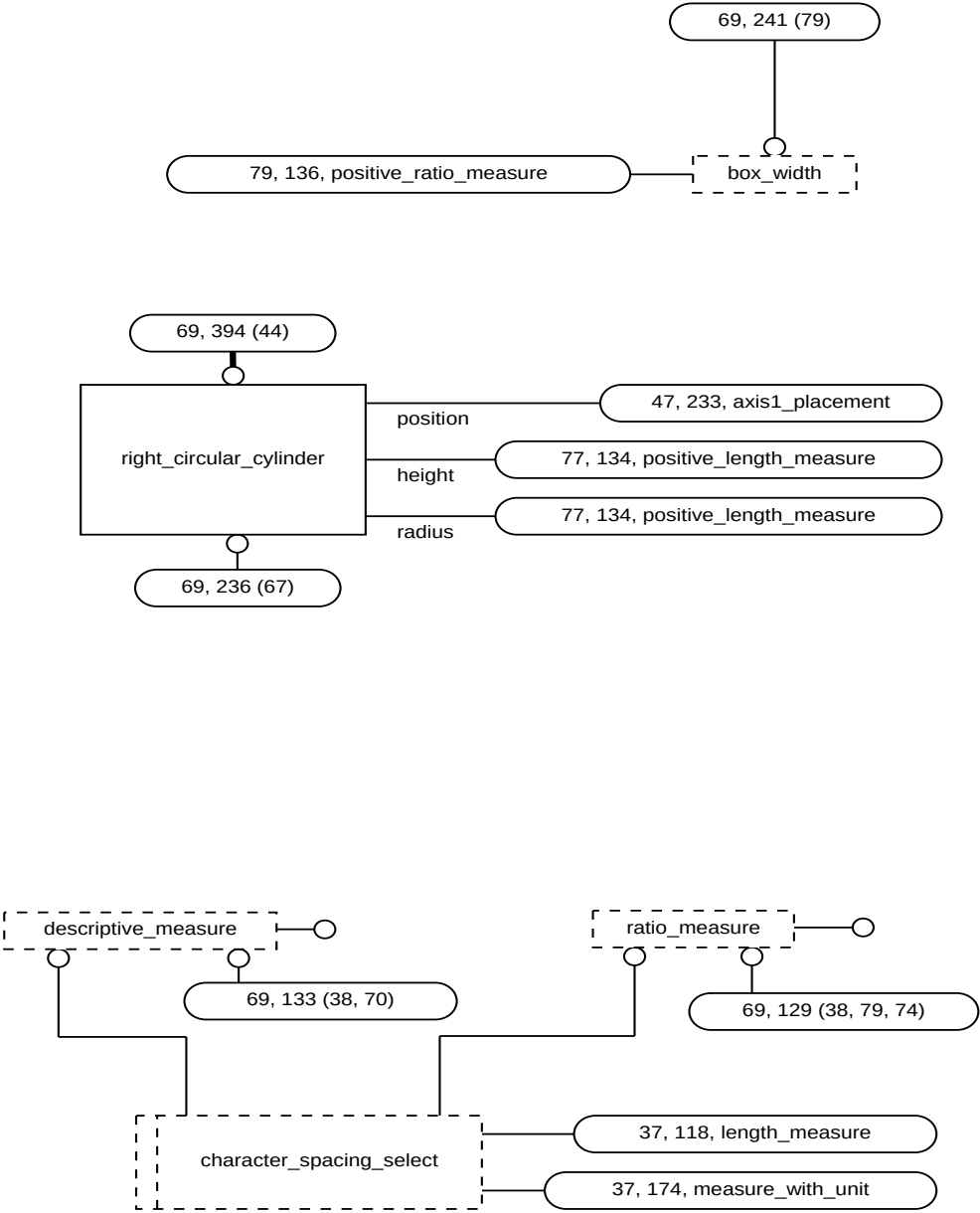


Figure J.69 – AIM EXPRESS-G diagram 69 of 80

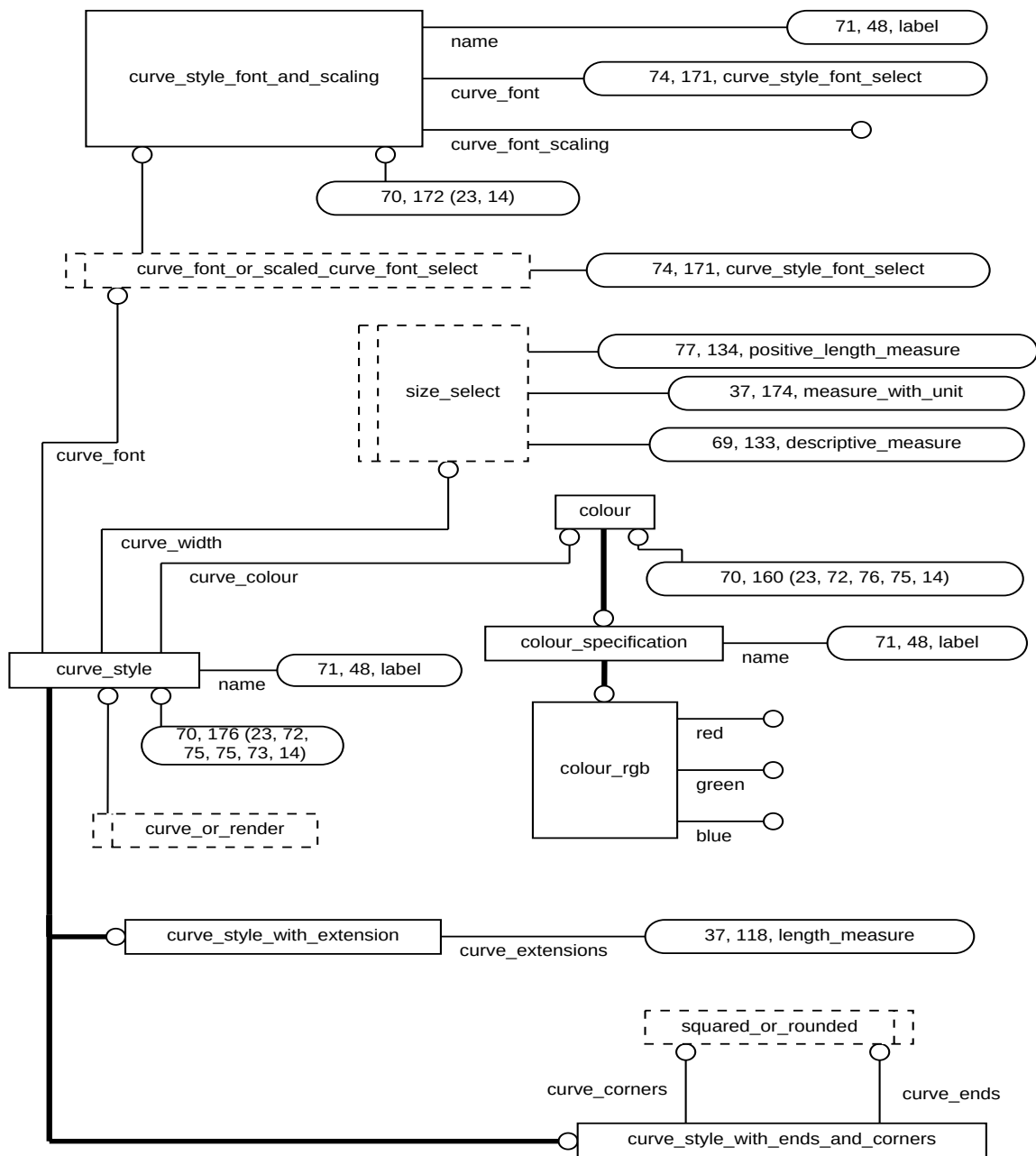


Figure J.70 – AIM EXPRESS-G diagram 70 of 80

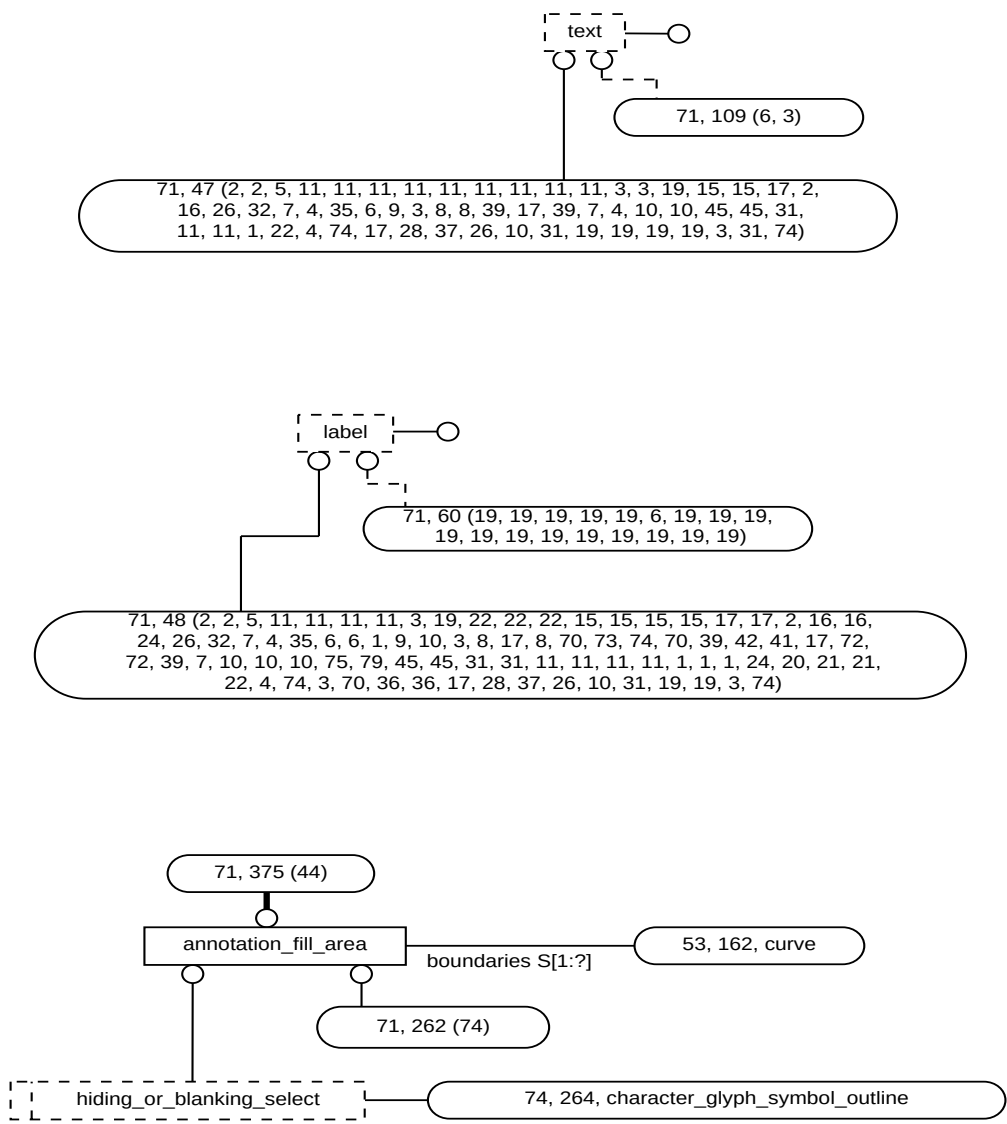


Figure J.71 – AIM EXPRESS-G diagram 71 of 80

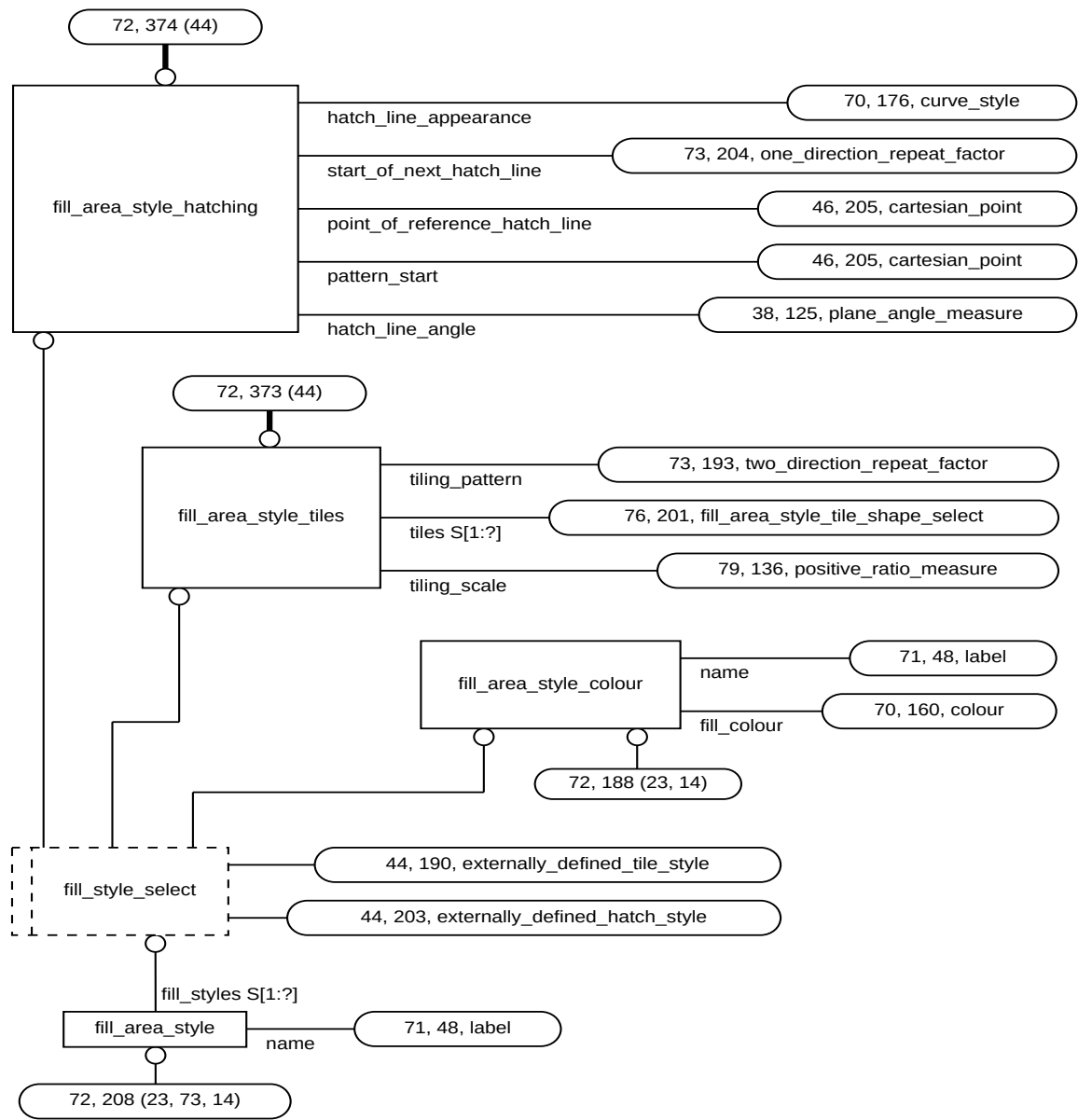


Figure J.72 – AIM EXPRESS-G diagram 72 of 80

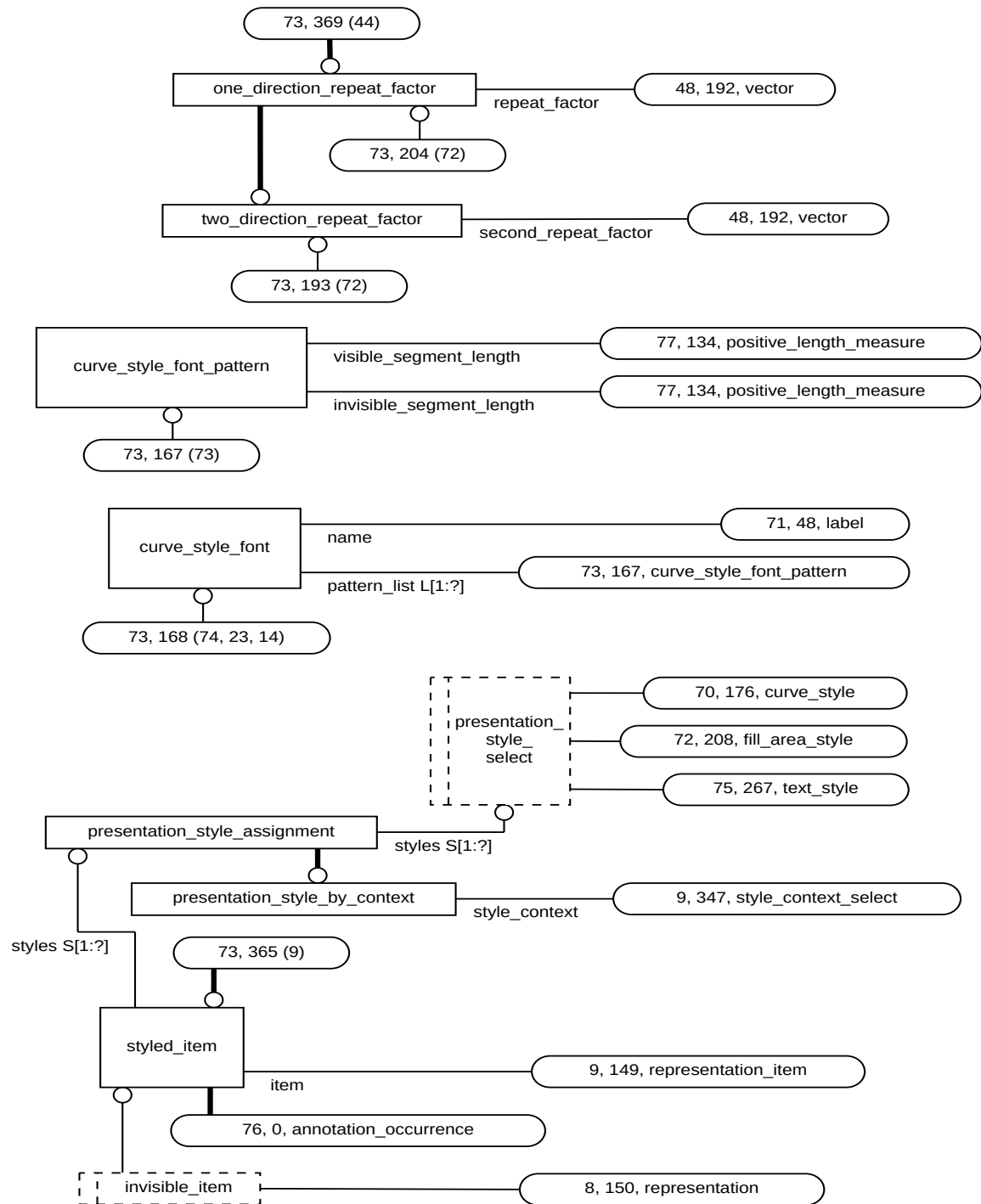


Figure J.73 – AIM EXPRESS-G diagram 73 of 80

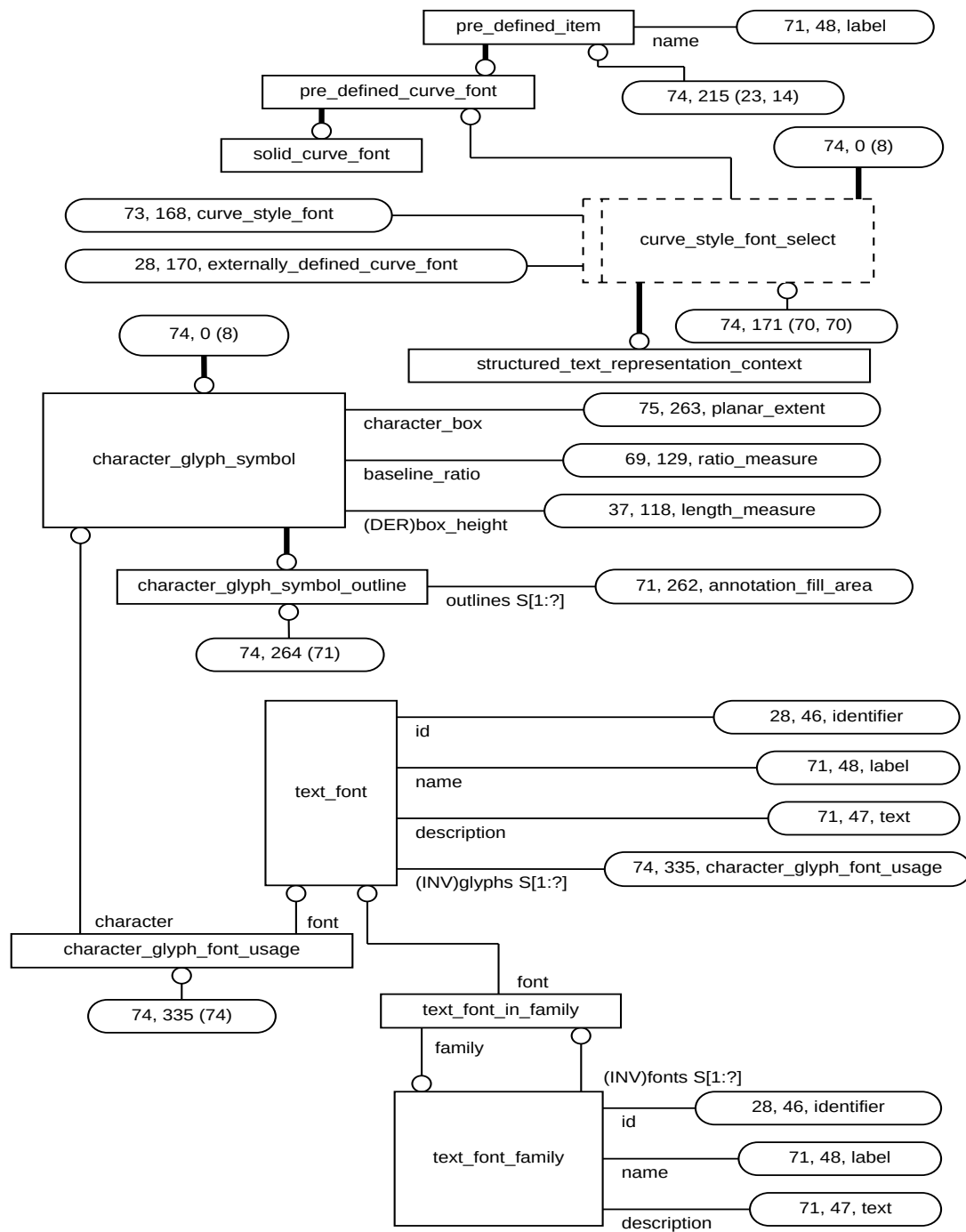


Figure J.74 – AIM EXPRESS-G diagram 74 of 80

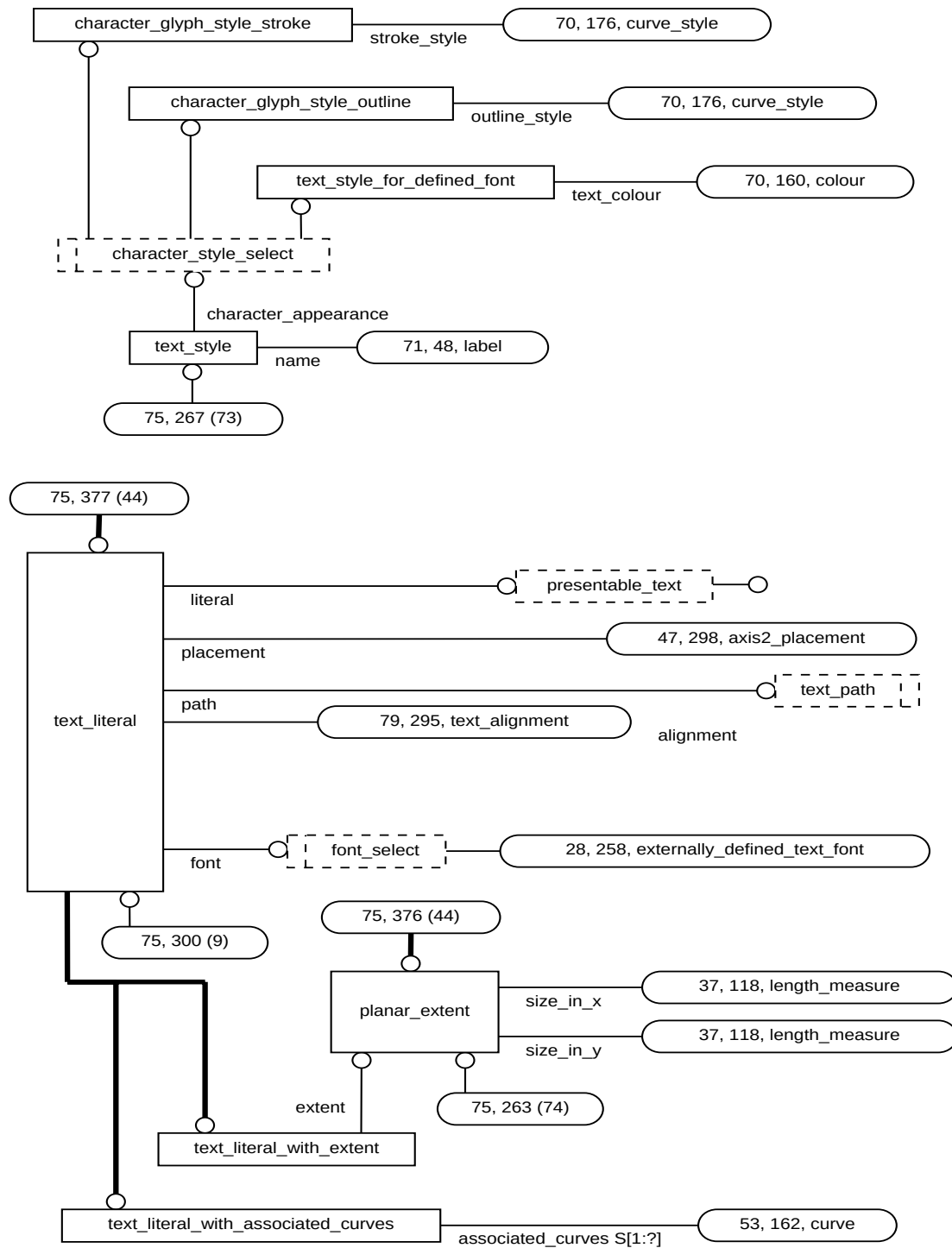


Figure J.75 – AIM EXPRESS-G diagram 75 of 80

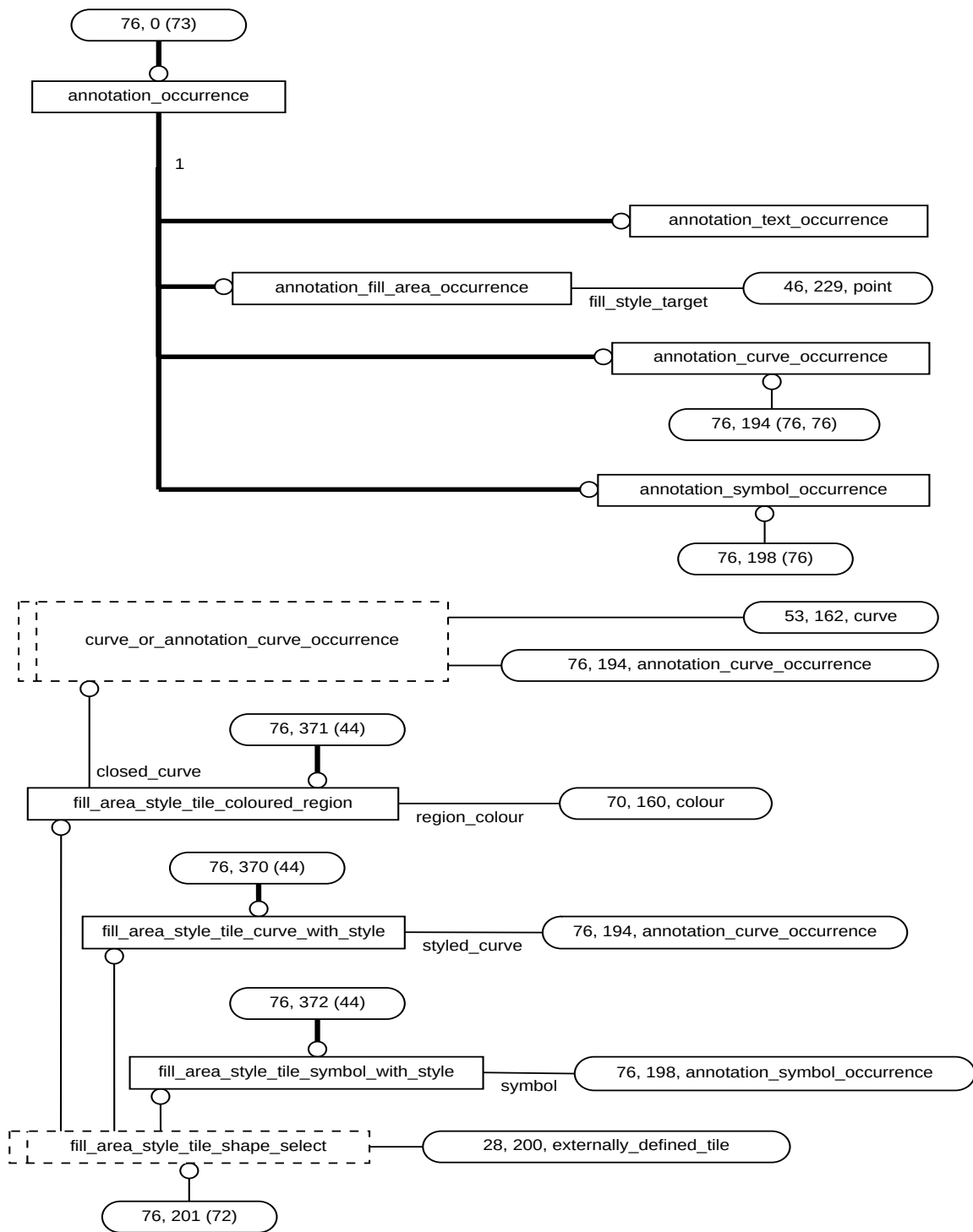


Figure J.76 – AIM EXPRESS-G diagram 76 of 80

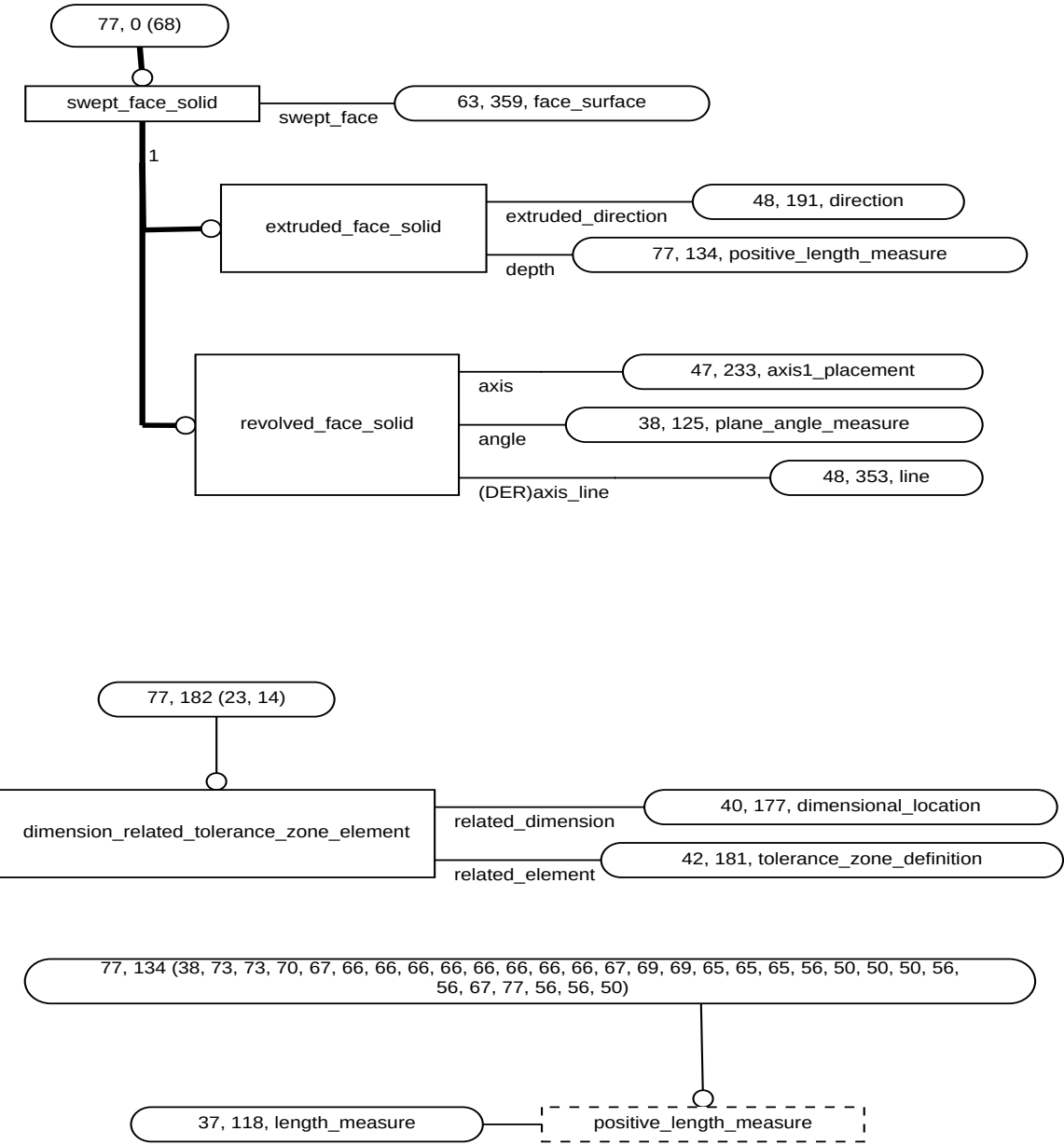


Figure J.77 – AIM EXPRESS-G diagram 77 of 80

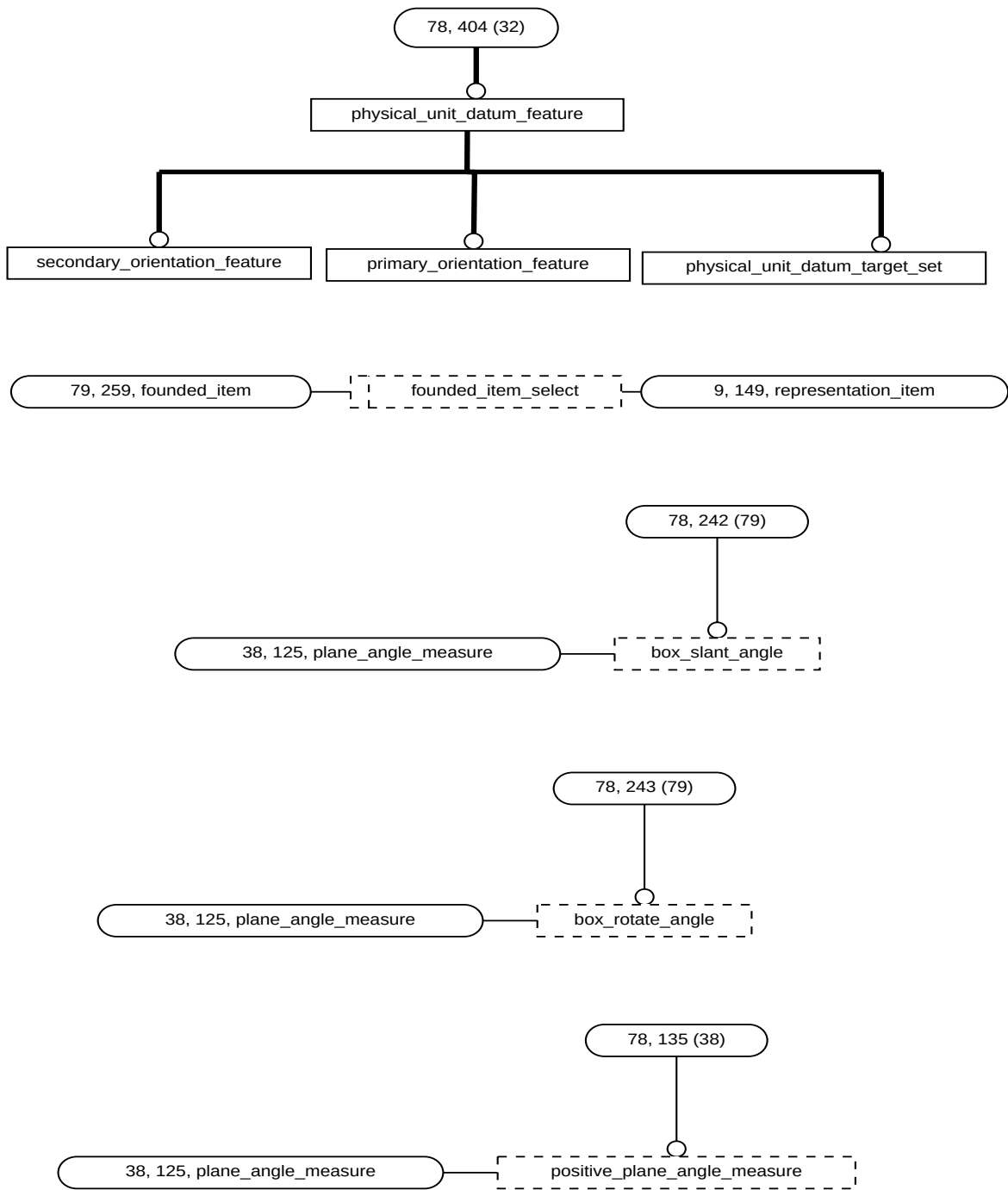


Figure J.78 – AIM EXPRESS-G diagram 78 of 80

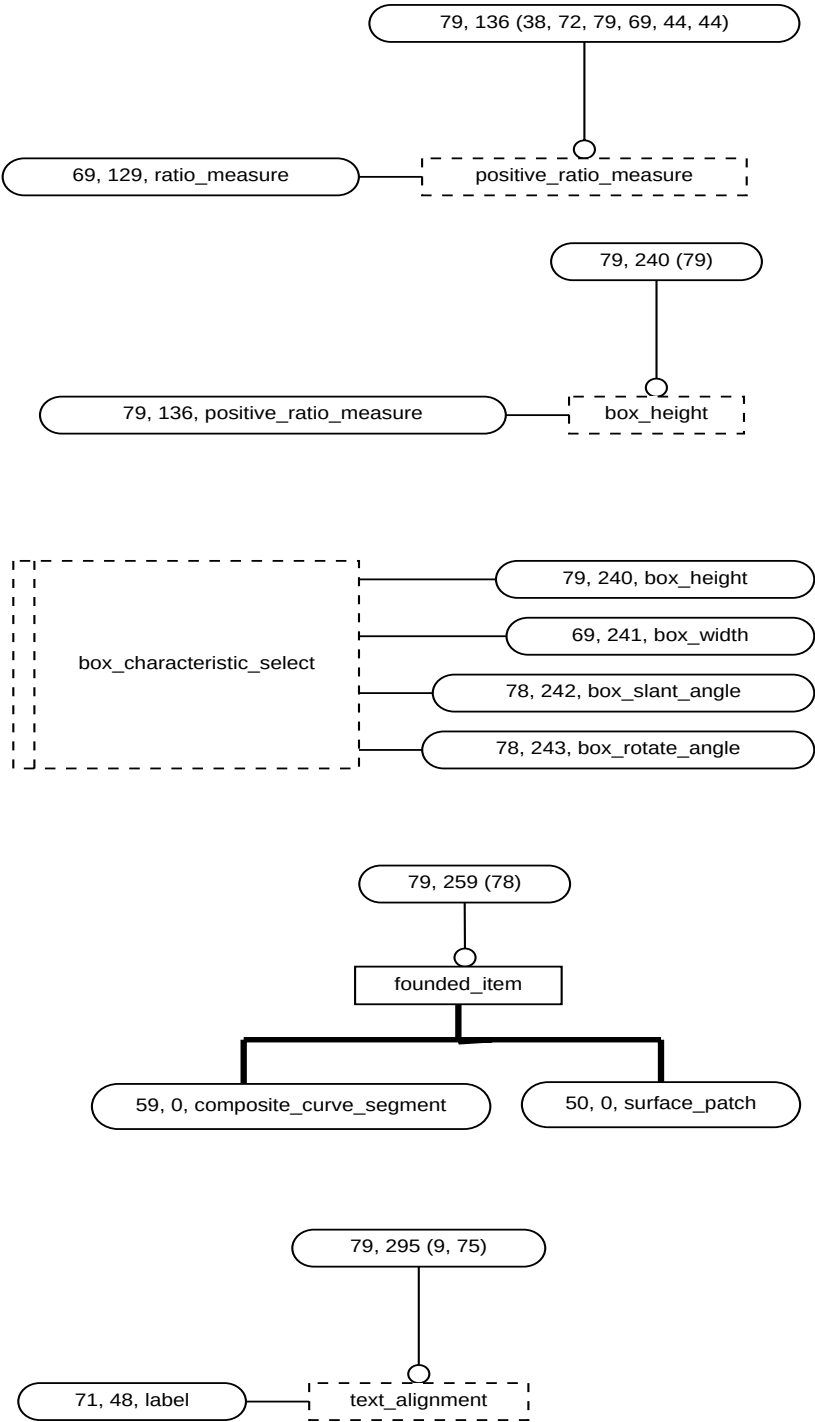


Figure J.79 – AIM EXPRESS-G diagram 79 of 80

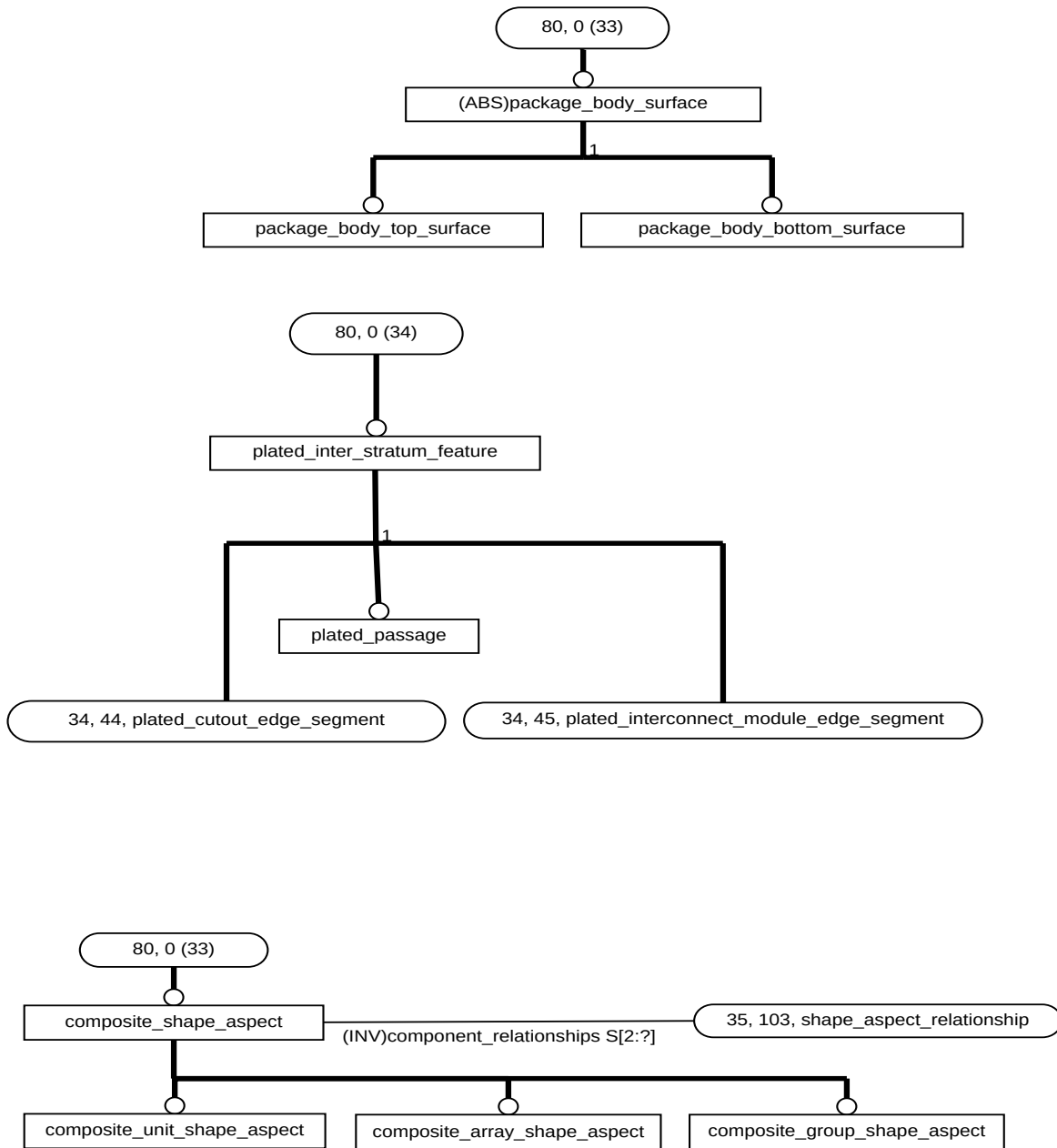


Figure J.80 – AIM EXPRESS-G diagram 80 of 80

Annex K
(informative)
AIM EXPRESS listing

This annex references a listing of the complete EXPRESS schema specified in Annex A of this part of ISO 10303 without comments or other explanatory text. It also references a listing of the EXPRESS entity names and corresponding short names as specified in Annex B of this part of ISO 10303. These listings are available in computer-interpretable form and can be found at the following URLs:

- EXPRESS: <http://www.mel.nist.gov/step/parts/part210/is/>
- Short names: <http://www.mel.nist.gov/div826/subject/apde/snr/>

If there is difficulty accessing these sites, contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex L (informative) Technical discussions

This annex provides two examples of how this application protocol might be used to represent a partial data set for a printed circuit assembly. The physical files are incomplete examples.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('IDF/AP210 AIM file'),'2;1');
FILE_NAME('aim210','2000-06-13T12:43:26-05:00',('Empilot Team'),
          ('PDES Inc. EMPilot'),'ST-DEVELOPER v7','IDF Version 2.0','');
FILE_SCHEMA(('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN
{iso standard 10303 part(210) version(1) object(1)
electronic-assembly-interconnect-and-packaging-design(1)}'));
ENDSEC;

DATA;

#10 = PRODUCT_RELATED_PRODUCT_CATEGORY('assembly module',$, (#1353));
#11 = PRODUCT_RELATED_PRODUCT_CATEGORY('interconnect module',$, (#1352));
#12 = APPLIED_GROUP_ASSIGNMENT(#1006, (#1408));
#13 = APPLIED_GROUP_ASSIGNMENT(#1007, (#1409));
#14 = ORGANIZATION_ROLE('document source');
#15 = ORGANIZATION_ROLE('document source');
#16 = APPLIED_ORGANIZATION_ASSIGNMENT(#828, #14, (#1344));
#17 = APPLIED_ORGANIZATION_ASSIGNMENT(#827, #15, (#1345));
#18 = APPLIED_DOCUMENT_REFERENCE(#1344, '', (#841, #1528, #1529));
#19 = APPLIED_DOCUMENT_REFERENCE(#1345, '', (#1527, #919, #920, #921, #922, #923,
#890, #891, #892, #893, #894, #1528, #1529));
#20 = DATE_TIME_ROLE('creation date');
#21 = DATE_TIME_ROLE('creation date');
#22 = DATE_TIME_ROLE('creation date');
#23 = DATE_TIME_ROLE('creation date');
#24 = DATE_TIME_ROLE('creation date');
#25 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#1422, #20, (#1345));
#26 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#1423, #21, (#1526));
#27 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#1423, #22, (#1525));
#28 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#1423, #23, (#1528));
#29 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#1423, #24, (#1529));
#30 = SHAPE_ASPECT_RELATIONSHIP('instantiated feature', '', #1011, #1018);
#31 = SHAPE_ASPECT_RELATIONSHIP('instantiated feature', '', #1012, #1019);
#32 = SHAPE_ASPECT_RELATIONSHIP('mounting surface', '', #1001, #1019);
#33 = SHAPE_ASPECT_RELATIONSHIP('mounting surface', '', #1002, #1018);
#34 = SHAPE_ASPECT_RELATIONSHIP('mounting surface', '', #1003, #1019);
#35 = SHAPE_ASPECT_RELATIONSHIP('mounting surface', '', #1004, #1018);
```

```

#36 = SHAPE_ASPECT_RELATIONSHIP('mounting surface', '', #1005, #1019);
#37 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #476, #485);
#38 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #477, #492);
#39 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #477, #493);
#40 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #478, #494);
#41 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #478, #495);
#42 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #477, #496);
#43 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #480, #487);
#44 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #481, #488);
#45 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #482, #489);
#46 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #483, #490);
#47 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #484, #491);
#48 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #479, #486);
#49 = SHAPE_ASPECT_RELATIONSHIP('via passage technology', '', #419, #476);
#50 = SHAPE_ASPECT_RELATIONSHIP('unsupported passage technology',
    '', #420, #477);
#51 = SHAPE_ASPECT_RELATIONSHIP('unsupported passage technology',
    '', #421, #478);
#52 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #422, #479);
#53 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #423, #480);
#54 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #424, #481);
#55 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #425, #482);
#56 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #426, #483);
#57 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #427, #484);
#58 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #419, #1013);
#59 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #420, #1013);
#60 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #421, #1013);
#61 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #422, #1013);
#62 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #423, #1013);
#63 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #424, #1014);
#64 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #425, #1015);
#65 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #426, #1016);

```

```

#66 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #427, #1017);
#67 = SHAPE_ASPECT_RELATIONSHIP('associated stratum extent', '', #1013, #227);
#68 = SHAPE_ASPECT_RELATIONSHIP('associated stratum extent', '', #1013, #228);
#69 = SHAPE_ASPECT_RELATIONSHIP('associated stratum extent', '', #1013, #229);
#70 = SHAPE_ASPECT_RELATIONSHIP('associated stratum extent', '', #1013, #230);
#71 = SHAPE_ASPECT_RELATIONSHIP('associated stratum extent', '', #1013, #231);
#72 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1082, #1082);
#73 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1083, #1083);
#74 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1075, #1075);
#75 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1087, #1087);
#76 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1088, #1088);
#77 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1089, #1089);
#78 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1090, #1090);
#79 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1086, #1086);
#80 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1084, #1084);
#81 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1084, #1084);
#82 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1085, #1085);
#83 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1085, #1085);
#84 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1084, #1084);
#85 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#86 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#87 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#88 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#89 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#90 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #1076, #1076);
#91 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',

```



```

    '',#1076,#1076);
#92 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1076,#1076);
#93 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1076,#1076);
#94 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1076,#1076);
#95 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1076,#1076);
#96 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1076,#1076);
#97 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1077,#1077);
#98 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#99 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1078,#1078);
#100 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1079,#1079);
#101 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#102 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#103 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#104 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#105 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#106 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#107 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1080,#1080);
#108 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1078,#1078);
#109 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1081,#1081);
#110 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '',#1081,#1081);
#111 = POLYLINE('',(#1161,#1162,#1163,#1164,#1165,#1166,#1167,#1168));
#112 = POLYLINE('',(#1175,#1176,#1177,#1178,#1179,#1180,#1181,#1182,#1183));
#113 = POLYLINE('',(#1187,#1188));
#114 = POLYLINE('',(#1192,#1193));
#115 = POLYLINE('',(#1197,#1198));
#116 = POLYLINE('',(#1202,#1203));
#117 = POLYLINE('',(#1207,#1208));

```

```

#118 = POLYLINE(' ', (#1212, #1213));
#119 = POLYLINE(' ', (#1217, #1218, #1219));
#120 = POLYLINE(' ', (#1220, #1221, #1222, #1223, #1224));
#121 = POLYLINE(' ', (#1231, #1232, #1233, #1234, #1235, #1236, #1237, #1238, #1239,
    #1240, #1241, #1242, #1243, #1244, #1245, #1246, #1247, #1248, #1249, #1250,
    #1251));
#122 = POLYLINE(' ', (#1252, #1253, #1254, #1255, #1256, #1257, #1258));
#123 = POLYLINE(' ', (#1259, #1260, #1261, #1262, #1263, #1264, #1265));
#124 = POLYLINE(' ', (#1266, #1267, #1268, #1269, #1270, #1271, #1272));
#125 = POLYLINE(' ', (#1273, #1274, #1275, #1276, #1277));
#126 = POLYLINE(' ', (#1278, #1279, #1280, #1281, #1282));
#127 = POLYLINE(' ', (#1283, #1284, #1285, #1286, #1287));
#128 = POLYLINE(' ', (#1288, #1289, #1290, #1291, #1292));
#129 = POLYLINE(' ', (#1293, #1294, #1295, #1296, #1297, #1298));
#130 = POLYLINE(' ', (#1299, #1300, #1301, #1302, #1303, #1304));
#131 = POLYLINE(' ', (#1305, #1306, #1307, #1308, #1309, #1310));
#132 = POLYLINE(' ', (#1311, #1312, #1313, #1314));
#133 = POLYLINE(' ', (#1318, #1319, #1320, #1321));
#134 = TRIMMED_CURVE(' ', #431, (#1158), (#1159), .T., .CARTESIAN.);
#135 = TRIMMED_CURVE(' ', #432, (#1169), (#1170), .F., .CARTESIAN.);
#136 = TRIMMED_CURVE(' ', #433, (#1172), (#1173), .F., .CARTESIAN.);
#137 = TRIMMED_CURVE(' ', #434, (#1184), (#1185), .F., .CARTESIAN.);
#138 = TRIMMED_CURVE(' ', #435, (#1189), (#1190), .F., .CARTESIAN.);
#139 = TRIMMED_CURVE(' ', #436, (#1194), (#1195), .F., .CARTESIAN.);
#140 = TRIMMED_CURVE(' ', #437, (#1199), (#1200), .F., .CARTESIAN.);
#141 = TRIMMED_CURVE(' ', #438, (#1204), (#1205), .F., .CARTESIAN.);
#142 = TRIMMED_CURVE(' ', #439, (#1209), (#1210), .F., .CARTESIAN.);
#143 = TRIMMED_CURVE(' ', #440, (#1214), (#1215), .F., .CARTESIAN.);
#144 = TRIMMED_CURVE(' ', #441, (#1225), (#1226), .F., .CARTESIAN.);
#145 = TRIMMED_CURVE(' ', #442, (#1228), (#1229), .F., .CARTESIAN.);
#146 = TRIMMED_CURVE(' ', #443, (#1315), (#1316), .T., .CARTESIAN.);
#147 = TRIMMED_CURVE(' ', #444, (#1322), (#1323), .T., .CARTESIAN.);
#148 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #134);
#149 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #111);
#150 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #135);
#151 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #136);
#152 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #112);
#153 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #137);
#154 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #113);
#155 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #138);
#156 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #114);
#157 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #139);
#158 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #115);
#159 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #140);
#160 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #116);
#161 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #141);

```

```

#162 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #117);
#163 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #142);
#164 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #118);
#165 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #143);
#166 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #119);
#167 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #120);
#168 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #144);
#169 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #145);
#170 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #121);
#171 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #122);
#172 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #123);
#173 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #124);
#174 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #125);
#175 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #126);
#176 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #127);
#177 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #128);
#178 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #129);
#179 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #130);
#180 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #131);
#181 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #132);
#182 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #146);
#183 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #133);
#184 = COMPOSITE_CURVE_SEGMENT( .CONTINUOUS., .T., #147);
#185 = COMPOSITE_CURVE( '', (#148, #149), .T.);
#186 = COMPOSITE_CURVE( '', (#150, #151), .T.);
#187 = COMPOSITE_CURVE( '', (#152), .T.);
#188 = COMPOSITE_CURVE( '', (#153, #154, #155, #156, #157, #158, #159, #160), .T.);
#189 = COMPOSITE_CURVE( '', (#161, #162, #163, #164, #165, #166), .T.);
#190 = COMPOSITE_CURVE( '', (#167), .T.);
#191 = COMPOSITE_CURVE( '', (#168, #169), .T.);
#192 = COMPOSITE_CURVE( '', (#170), .T.);
#193 = COMPOSITE_CURVE( '', (#171), .T.);
#194 = COMPOSITE_CURVE( '', (#172), .T.);
#195 = COMPOSITE_CURVE( '', (#173), .T.);
#196 = COMPOSITE_CURVE( '', (#174), .T.);
#197 = COMPOSITE_CURVE( '', (#175), .T.);
#198 = COMPOSITE_CURVE( '', (#176), .T.);
#199 = COMPOSITE_CURVE( '', (#177), .T.);
#200 = COMPOSITE_CURVE( '', (#178), .T.);
#201 = COMPOSITE_CURVE( '', (#179), .T.);
#202 = COMPOSITE_CURVE( '', (#180), .T.);
#203 = COMPOSITE_CURVE( '', (#181, #182, #183, #184), .T.);
#204 = INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY( '',
    'inter stratum feature category');
#205 = INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY( '',
    'stratum feature category');

```

```

#206 = INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY('',
    'inter stratum feature category');
#207 = INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY('',
    'stratum feature category');
#208 = INTERCONNECT_MODULE_DESIGN_OBJECT_CATEGORY('',
    'via category');
#209 = PROPERTY_DEFINITION_RELATIONSHIP('constrained object','',#594,#595);
#210 = PROPERTY_DEFINITION_RELATIONSHIP('constrained object','',#598,#599);
#211 = PROPERTY_DEFINITION_RELATIONSHIP('constrained object','',#602,#603);
#212 = PROPERTY_DEFINITION_RELATIONSHIP('constrained object','',#606,#607);
#213 = PROPERTY_DEFINITION_RELATIONSHIP('constrained object','',#610,#611);
#214 = PROPERTY_DEFINITION_RELATIONSHIP('technology usage','',#631,#630);
#215 = PROPERTY_DEFINITION_RELATIONSHIP('technology usage','',#633,#632);
#216 = PROPERTY_DEFINITION_RELATIONSHIP('technology usage','',#635,#634);
#217 = PROPERTY_DEFINITION_RELATIONSHIP('requirement','',#1411,#676);
#218 = PROPERTY_DEFINITION_RELATIONSHIP('requirement','',#1411,#677);
#219 = PROPERTY_DEFINITION_RELATIONSHIP('requirement','',#1412,#678);
#220 = PROPERTY_DEFINITION_RELATIONSHIP('requirement','',#1412,#679);
#221 = PROPERTY_DEFINITION_RELATIONSHIP('requirement','',#1413,#680);
#222 = GROUP_PRODUCT_DEFINITION('','',#1371,#1726);
#223 = GROUP_PRODUCT_DEFINITION('','',#1371,#1726);
#224 = GROUP_PRODUCT_DEFINITION('','',#1371,#1726);
#225 = GROUP_PRODUCT_DEFINITION('','',#1371,#1726);
#226 = GROUP_PRODUCT_DEFINITION('','',#1371,#1726);
#227 = GROUP_SHAPE_ASPECT('','interconnect module constraint region',
    #1628,.F.);
#228 = GROUP_SHAPE_ASPECT('','interconnect module constraint region',
    #1629,.F.);
#229 = GROUP_SHAPE_ASPECT('','interconnect module constraint region',
    #1630,.F.);
#230 = GROUP_SHAPE_ASPECT('','interconnect module constraint region',
    #1631,.F.);
#231 = GROUP_SHAPE_ASPECT('','interconnect module constraint region',
    #1632,.F.);
#232 = REPRESENTATION_MAP(#1492,#1075);
#233 = REPRESENTATION_MAP(#1493,#1076);
#234 = REPRESENTATION_MAP(#1493,#1076);
#235 = REPRESENTATION_MAP(#1493,#1076);
#236 = REPRESENTATION_MAP(#1493,#1076);
#237 = REPRESENTATION_MAP(#1493,#1076);
#238 = REPRESENTATION_MAP(#1493,#1076);
#239 = REPRESENTATION_MAP(#1493,#1076);
#240 = REPRESENTATION_MAP(#1493,#1076);
#241 = REPRESENTATION_MAP(#1493,#1076);
#242 = REPRESENTATION_MAP(#1493,#1076);
#243 = REPRESENTATION_MAP(#1494,#1077);

```

```

#244 = REPRESENTATION_MAP(#1497,#1080);
#245 = REPRESENTATION_MAP(#1495,#1078);
#246 = REPRESENTATION_MAP(#1496,#1079);
#247 = REPRESENTATION_MAP(#1497,#1080);
#248 = REPRESENTATION_MAP(#1497,#1080);
#249 = REPRESENTATION_MAP(#1495,#1078);
#250 = REPRESENTATION_MAP(#1498,#1081);
#251 = REPRESENTATION_MAP(#1498,#1081);
#252 = REPRESENTATION_MAP(#1499,#1082);
#253 = REPRESENTATION_MAP(#1500,#1083);
#254 = REPRESENTATION_MAP(#1507,#1087);
#255 = REPRESENTATION_MAP(#1508,#1088);
#256 = REPRESENTATION_MAP(#1509,#1089);
#257 = REPRESENTATION_MAP(#1510,#1090);
#258 = REPRESENTATION_MAP(#1505,#1086);
#259 = REPRESENTATION_MAP(#1501,#1084);
#260 = REPRESENTATION_MAP(#1501,#1084);
#261 = REPRESENTATION_MAP(#1503,#1085);
#262 = REPRESENTATION_MAP(#1503,#1085);
#263 = REPRESENTATION_MAP(#1501,#1084);
#264 = REPRESENTATION_MAP(#1483,#1066);
#265 = REPRESENTATION_MAP(#1484,#1067);
#266 = REPRESENTATION_MAP(#1485,#1068);
#267 = REPRESENTATION_MAP(#1486,#1069);
#268 = REPRESENTATION_MAP(#1487,#1070);
#269 = REPRESENTATION_MAP(#1488,#1071);
#270 = REPRESENTATION_MAP(#1489,#1072);
#271 = COMPONENT_LOCATION('planar projected shape',(#1680,#303,#869),#837);
#272 = COMPONENT_LOCATION('planar projected shape',(#1681,#304,#870),#837);
#273 = COMPONENT_LOCATION('planar projected shape',(#1682,#305,#849),#837);
#274 = COMPONENT_LOCATION('planar projected shape',(#1683,#306,#871),#837);
#275 = COMPONENT_LOCATION('planar projected shape',(#1684,#307,#872),#837);
#276 = COMPONENT_LOCATION('planar projected shape',(#1685,#308,#873),#837);
#277 = COMPONENT_LOCATION('planar projected shape',(#1686,#309,#874),#837);
#278 = COMPONENT_LOCATION('planar projected shape',(#1687,#310,#850),#837);
#279 = COMPONENT_LOCATION('planar projected shape',(#1688,#311,#851),#837);
#280 = COMPONENT_LOCATION('planar projected shape',(#1689,#312,#852),#837);
#281 = COMPONENT_LOCATION('planar projected shape',(#1690,#313,#853),#837);
#282 = COMPONENT_LOCATION('planar projected shape',(#1691,#314,#854),#837);
#283 = COMPONENT_LOCATION('planar projected shape',(#1692,#315,#855),#837);
#284 = COMPONENT_LOCATION('planar projected shape',(#1693,#316,#856),#837);
#285 = COMPONENT_LOCATION('planar projected shape',(#1694,#317,#857),#837);
#286 = COMPONENT_LOCATION('planar projected shape',(#1695,#318,#858),#837);
#287 = COMPONENT_LOCATION('planar projected shape',(#1696,#319,#859),#837);
#288 = COMPONENT_LOCATION('planar projected shape',(#1697,#320,#860),#837);
#289 = COMPONENT_LOCATION('planar projected shape',(#1698,#321,#861),#837);

```

```

#290 = COMPONENT_LOCATION('planar projected shape', (#1699, #322, #862), #837);
#291 = COMPONENT_LOCATION('planar projected shape', (#1700, #323, #863), #837);
#292 = COMPONENT_LOCATION('planar projected shape', (#1701, #324, #864), #837);
#293 = COMPONENT_LOCATION('planar projected shape', (#1702, #325, #865), #837);
#294 = COMPONENT_LOCATION('planar projected shape', (#1703, #326, #866), #837);
#295 = COMPONENT_LOCATION('planar projected shape', (#1704, #327, #867), #837);
#296 = COMPONENT_LOCATION('planar projected shape', (#1705, #328, #868), #837);
#297 = COMPONENT_LOCATION('planar projected shape', (#1706, #329, #875), #837);
#298 = COMPONENT_LOCATION('planar projected shape', (#1707, #330, #876), #837);
#299 = COMPONENT_LOCATION('planar projected shape', (#1708, #331, #877), #837);
#300 = COMPONENT_LOCATION('planar projected shape', (#1709, #332, #878), #837);
#301 = COMPONENT_LOCATION('planar projected shape', (#1710, #333, #879), #837);
#302 = COMPONENT_LOCATION('planar projected shape', (#1711, #334, #880), #837);
#303 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #924, #976, #1125, 1.);
#304 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #925, #977, #1126, 1.);
#305 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #926, #978, #1127, 1.);
#306 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #928, #979, #1129, 1.);
#307 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #929, #980, #1130, 1.);
#308 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #930, #981, #1131, 1.);
#309 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #931, #982, #1132, 1.);
#310 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #932, #983, #1133, 1.);
#311 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #933, #984, #1134, 1.);
#312 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #934, #985, #1135, 1.);
#313 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #935, #986, #1136, 1.);
#314 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #936, #969, #1137, 1.);
#315 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #937, #970, #1138, 1.);
#316 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #938, #971, #1139, 1.);
#317 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #939, #987, #1140, 1.);
#318 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #940, #988, #1141, 1.);
#319 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #941, #989, #1142, 1.);
#320 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #942, #990, #1143, 1.);
#321 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #943, #991, #1144, 1.);
#322 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #944, #972, #1145, 1.);
#323 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #945, #973, #1146, 1.);
#324 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #946, #992, #1147, 1.);
#325 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #947, #993, #1148, 1.);
#326 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #948, #974, #1149, 1.);
#327 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #949, #975, #1150, 1.);
#328 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #950, #994, #1151, 1.);
#329 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #951, #995, #1152, 1.);
#330 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #952, #996, #1153, 1.);
#331 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #953, #997, #1154, 1.);
#332 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #954, #998, #1155, 1.);
#333 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #955, #999, #1156, 1.);
#334 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #956, #1000, #1157, 1.);
#335 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('', '', '', #957, #958, #1325, 1.);

```

```

#336 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',' ',#959,#960,#1326,1.);
#337 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',' ',#961,#962,#1327,1.);
#338 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',' ',#963,#964,#1328,1.);
#339 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',' ',#965,#966,#1329,1.);
#340 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',' ',#967,#968,#1330,1.);
#341 = SHAPE_DEFINITION_REPRESENTATION(#1566,#1082);
#342 = SHAPE_DEFINITION_REPRESENTATION(#1567,#1083);
#343 = SHAPE_DEFINITION_REPRESENTATION(#1637,#1075);
#344 = SHAPE_DEFINITION_REPRESENTATION(#1568,#1087);
#345 = SHAPE_DEFINITION_REPRESENTATION(#1569,#1088);
#346 = SHAPE_DEFINITION_REPRESENTATION(#1570,#1089);
#347 = SHAPE_DEFINITION_REPRESENTATION(#1571,#1090);
#348 = SHAPE_DEFINITION_REPRESENTATION(#1577,#1086);
#349 = SHAPE_DEFINITION_REPRESENTATION(#1572,#1084);
#350 = SHAPE_DEFINITION_REPRESENTATION(#1573,#1084);
#351 = SHAPE_DEFINITION_REPRESENTATION(#1574,#1085);
#352 = SHAPE_DEFINITION_REPRESENTATION(#1575,#1085);
#353 = SHAPE_DEFINITION_REPRESENTATION(#1576,#1084);
#354 = SHAPE_DEFINITION_REPRESENTATION(#1533,#1076);
#355 = SHAPE_DEFINITION_REPRESENTATION(#1534,#1076);
#356 = SHAPE_DEFINITION_REPRESENTATION(#1535,#1076);
#357 = SHAPE_DEFINITION_REPRESENTATION(#1536,#1076);
#358 = SHAPE_DEFINITION_REPRESENTATION(#1537,#1076);
#359 = SHAPE_DEFINITION_REPRESENTATION(#1538,#1076);
#360 = SHAPE_DEFINITION_REPRESENTATION(#1539,#1076);
#361 = SHAPE_DEFINITION_REPRESENTATION(#1540,#1076);
#362 = SHAPE_DEFINITION_REPRESENTATION(#1541,#1076);
#363 = SHAPE_DEFINITION_REPRESENTATION(#1542,#1076);
#364 = SHAPE_DEFINITION_REPRESENTATION(#1543,#1076);
#365 = SHAPE_DEFINITION_REPRESENTATION(#1544,#1076);
#366 = SHAPE_DEFINITION_REPRESENTATION(#1545,#1077);
#367 = SHAPE_DEFINITION_REPRESENTATION(#1546,#1080);
#368 = SHAPE_DEFINITION_REPRESENTATION(#1547,#1078);
#369 = SHAPE_DEFINITION_REPRESENTATION(#1548,#1079);
#370 = SHAPE_DEFINITION_REPRESENTATION(#1549,#1080);
#371 = SHAPE_DEFINITION_REPRESENTATION(#1550,#1080);
#372 = SHAPE_DEFINITION_REPRESENTATION(#1551,#1080);
#373 = SHAPE_DEFINITION_REPRESENTATION(#1552,#1080);
#374 = SHAPE_DEFINITION_REPRESENTATION(#1553,#1080);
#375 = SHAPE_DEFINITION_REPRESENTATION(#1554,#1080);
#376 = SHAPE_DEFINITION_REPRESENTATION(#1555,#1080);
#377 = SHAPE_DEFINITION_REPRESENTATION(#1556,#1078);
#378 = SHAPE_DEFINITION_REPRESENTATION(#1530,#1081);
#379 = SHAPE_DEFINITION_REPRESENTATION(#1531,#1081);
#380 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('','assembly composition',
    '',#1529,#1648,$);

```

```
#381 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1529, #1649, $);  
#382 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #1647, $);  
#383 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1529, #1650, $);  
#384 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1529, #1651, $);  
#385 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1529, #1652, $);  
#386 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1529, #1653, $);  
#387 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #895, 'C1');  
#388 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #896, 'C10');  
#389 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #897, 'C11');  
#390 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #898, 'C2');  
#391 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #899, 'C3');  
#392 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #900, 'C4');  
#393 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #901, 'C5');  
#394 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #902, 'C6');  
#395 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #903, 'C60');  
#396 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #904, 'C7');  
#397 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #905, 'C8');  
#398 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #906, 'C9');  
#399 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #907, 'P1');  
#400 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #908, 'U1');  
#401 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #909, 'U10');  
#402 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #910, 'U11');  
#403 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',  
    '', #1528, #911, 'U2');
```



```

#404 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #912, 'U3');
#405 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #913, 'U4');
#406 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #914, 'U5');
#407 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #915, 'U6');
#408 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #916, 'U7');
#409 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #917, 'U8');
#410 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #918, 'U9');
#411 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #1645, 'norefdes1');
#412 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1528, #1646, 'norefdes2');
#413 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1659, $);
#414 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1654, $);
#415 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1655, $);
#416 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1656, $);
#417 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1657, $);
#418 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #1529, #1658, $);
#419 = PASSAGE_TECHNOLOGY('150 mil plated hole',
    'default via definition', #1587, .F.);
#420 = PASSAGE_TECHNOLOGY('61 mil non-plated hole',
    'default unsupported passage definition', #1588, .F.);
#421 = PASSAGE_TECHNOLOGY('100 mil non-plated hole',
    'default unsupported passage definition', #1589, .F.);
#422 = PASSAGE_TECHNOLOGY('interconnect module edge', '', #1590, .F.);
#423 = PASSAGE_TECHNOLOGY('cutout a', '', #1591, .F.);
#424 = PASSAGE_TECHNOLOGY('cutout b', '', #1592, .F.);
#425 = PASSAGE_TECHNOLOGY('cutout c', '', #1593, .F.);
#426 = PASSAGE_TECHNOLOGY('cutout d', '', #1594, .F.);
#427 = PASSAGE_TECHNOLOGY('cutout e', '', #1595, .F.);
#428 = CIRCLE('', #1502, 30.5);
#429 = CIRCLE('', #1504, 50.);
#430 = CIRCLE('', #1506, 75.);
#431 = CIRCLE('', #1511, 250.);

```

```

#432 = CIRCLE(' ', #1512, 223.6);
#433 = CIRCLE(' ', #1513, 223.6);
#434 = CIRCLE(' ', #1514, 50.);
#435 = CIRCLE(' ', #1515, 50.);
#436 = CIRCLE(' ', #1516, 50.);
#437 = CIRCLE(' ', #1517, 50.);
#438 = CIRCLE(' ', #1518, 120.);
#439 = CIRCLE(' ', #1519, 120.);
#440 = CIRCLE(' ', #1520, 316.296975896509);
#441 = CIRCLE(' ', #1521, 447.2);
#442 = CIRCLE(' ', #1522, 447.2);
#443 = CIRCLE(' ', #1523, 50.);
#444 = CIRCLE(' ', #1524, 200.);
#445 = GEOMETRIC_CURVE_SET(' ', (#428));
#446 = GEOMETRIC_CURVE_SET(' ', (#429));
#447 = GEOMETRIC_CURVE_SET(' ', (#430));
#448 = GEOMETRIC_CURVE_SET(' ', (#185));
#449 = GEOMETRIC_CURVE_SET(' ', (#186));
#450 = GEOMETRIC_CURVE_SET(' ', (#187));
#451 = GEOMETRIC_CURVE_SET(' ', (#188));
#452 = GEOMETRIC_CURVE_SET(' ', (#189));
#453 = GEOMETRIC_CURVE_SET(' ', (#190));
#454 = GEOMETRIC_CURVE_SET(' ', (#191));
#455 = GEOMETRIC_CURVE_SET(' ', (#192));
#456 = GEOMETRIC_CURVE_SET(' ', (#193));
#457 = GEOMETRIC_CURVE_SET(' ', (#194));
#458 = GEOMETRIC_CURVE_SET(' ', (#195));
#459 = GEOMETRIC_CURVE_SET(' ', (#196));
#460 = GEOMETRIC_CURVE_SET(' ', (#197));
#461 = GEOMETRIC_CURVE_SET(' ', (#198));
#462 = GEOMETRIC_CURVE_SET(' ', (#199));
#463 = GEOMETRIC_CURVE_SET(' ', (#200));
#464 = GEOMETRIC_CURVE_SET(' ', (#201));
#465 = GEOMETRIC_CURVE_SET(' ', (#202));
#466 = GEOMETRIC_CURVE_SET(' ', (#203));
#467 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#468 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#469 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#470 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#471 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#472 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#473 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#474 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#475 = EXTERNAL_SOURCE(IDENTIFIER(' '));
#476 = PART_TEMPLATE_DEFINITION(' ', 'via template', #1578,
    .F., IDENTIFIER(' '), #467);

```

```

#477 = PART_TEMPLATE_DEFINITION('', 'unsupported passage template', #1579,
    .F., IDENTIFIER(''), #468);
#478 = PART_TEMPLATE_DEFINITION('', 'unsupported passage template', #1580,
    .F., IDENTIFIER(''), #469);
#479 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1581,
    .F., IDENTIFIER(''), #470);
#480 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1582,
    .F., IDENTIFIER(''), #471);
#481 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1583,
    .F., IDENTIFIER(''), #472);
#482 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1584,
    .F., IDENTIFIER(''), #473);
#483 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1585,
    .F., IDENTIFIER(''), #474);
#484 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template', #1586,
    .F., IDENTIFIER(''), #475);
#485 = PLATED_PASSAGE('', 'interfacial connection', #1577, .F.);
#486 = INTER_STRATUM_FEATURE('', 'interconnect module edge', #1566, .F.);
#487 = INTER_STRATUM_FEATURE('', 'cutout', #1567, .F.);
#488 = INTER_STRATUM_FEATURE('', 'cutout', #1568, .F.);
#489 = INTER_STRATUM_FEATURE('', 'cutout', #1569, .F.);
#490 = INTER_STRATUM_FEATURE('', 'cutout', #1570, .F.);
#491 = INTER_STRATUM_FEATURE('', 'cutout', #1571, .F.);
#492 = INTER_STRATUM_FEATURE('', 'unsupported passage', #1572, .F.);
#493 = INTER_STRATUM_FEATURE('', 'unsupported passage', #1573, .F.);
#494 = INTER_STRATUM_FEATURE('', 'unsupported passage', #1574, .F.);
#495 = INTER_STRATUM_FEATURE('', 'unsupported passage', #1575, .F.);
#496 = INTER_STRATUM_FEATURE('', 'unsupported passage', #1576, .F.);
#497 = REPRESENTATION_RELATIONSHIP('', '', #1073, #1435);
#498 = REPRESENTATION_RELATIONSHIP('', '', #1073, #1436);
#499 = REPRESENTATION_RELATIONSHIP('', '', #1073, #1437);
#500 = REPRESENTATION_RELATIONSHIP('', '', #1074, #1435);
#501 = REPRESENTATION_RELATIONSHIP('', '', #1074, #1436);
#502 = REPRESENTATION_RELATIONSHIP('', '', #1074, #1437);
#503 = REPRESENTATION_RELATIONSHIP('', '', #1075, #1435);
#504 = REPRESENTATION_RELATIONSHIP('', '', #1075, #1436);
#505 = REPRESENTATION_RELATIONSHIP('', '', #1075, #1437);
#506 = REPRESENTATION_RELATIONSHIP('', '', #1076, #1435);
#507 = REPRESENTATION_RELATIONSHIP('', '', #1076, #1436);
#508 = REPRESENTATION_RELATIONSHIP('', '', #1076, #1438);
#509 = REPRESENTATION_RELATIONSHIP('', '', #1077, #1435);
#510 = REPRESENTATION_RELATIONSHIP('', '', #1077, #1436);
#511 = REPRESENTATION_RELATIONSHIP('', '', #1077, #1438);
#512 = REPRESENTATION_RELATIONSHIP('', '', #1078, #1435);
#513 = REPRESENTATION_RELATIONSHIP('', '', #1078, #1436);
#514 = REPRESENTATION_RELATIONSHIP('', '', #1078, #1438);

```

```

#515 = REPRESENTATION_RELATIONSHIP('', '', #1079, #1435);
#516 = REPRESENTATION_RELATIONSHIP('', '', #1079, #1436);
#517 = REPRESENTATION_RELATIONSHIP('', '', #1079, #1438);
#518 = REPRESENTATION_RELATIONSHIP('', '', #1080, #1435);
#519 = REPRESENTATION_RELATIONSHIP('', '', #1080, #1436);
#520 = REPRESENTATION_RELATIONSHIP('', '', #1080, #1438);
#521 = REPRESENTATION_RELATIONSHIP('', '', #1081, #1435);
#522 = REPRESENTATION_RELATIONSHIP('', '', #1081, #1436);
#523 = REPRESENTATION_RELATIONSHIP('', '', #1081, #1438);
#524 = REPRESENTATION_RELATIONSHIP('', '', #1082, #1435);
#525 = REPRESENTATION_RELATIONSHIP('', '', #1082, #1436);
#526 = REPRESENTATION_RELATIONSHIP('', '', #1083, #1451);
#527 = REPRESENTATION_RELATIONSHIP('', '', #1083, #1436);
#528 = REPRESENTATION_RELATIONSHIP('', '', #1084, #1435);
#529 = REPRESENTATION_RELATIONSHIP('', '', #1084, #1436);
#530 = REPRESENTATION_RELATIONSHIP('', '', #1085, #1435);
#531 = REPRESENTATION_RELATIONSHIP('', '', #1085, #1436);
#532 = REPRESENTATION_RELATIONSHIP('', '', #1086, #1435);
#533 = REPRESENTATION_RELATIONSHIP('', '', #1086, #1436);
#534 = REPRESENTATION_RELATIONSHIP('', '', #1087, #1451);
#535 = REPRESENTATION_RELATIONSHIP('', '', #1087, #1436);
#536 = REPRESENTATION_RELATIONSHIP('', '', #1088, #1451);
#537 = REPRESENTATION_RELATIONSHIP('', '', #1088, #1436);
#538 = REPRESENTATION_RELATIONSHIP('', '', #1089, #1451);
#539 = REPRESENTATION_RELATIONSHIP('', '', #1089, #1436);
#540 = REPRESENTATION_RELATIONSHIP('', '', #1090, #1451);
#541 = REPRESENTATION_RELATIONSHIP('', '', #1090, #1436);
#542 = REPRESENTATION_RELATIONSHIP('', '', #1424, #1427);
#543 = REPRESENTATION_RELATIONSHIP('', '', #1424, #1428);
#544 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #842, #843);
#545 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #843, #844);
#546 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #844, #845);
#547 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(62.), #830)
    REPRESENTATION_ITEM('maximum thickness'));
#548 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(62.), #830)
    REPRESENTATION_ITEM('minimum thickness'));
#549 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(250.), #830)

```

```

        REPRESENTATION_ITEM('measure'));
#550 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.),#830)
        REPRESENTATION_ITEM('measure'));
#551 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(67.),#830)
        REPRESENTATION_ITEM('maximum body height above seating plane'));
#552 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(180.),#830)
        REPRESENTATION_ITEM('maximum body height above seating plane'));
#553 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(180.),#830)
        REPRESENTATION_ITEM('maximum body height above seating plane'));
#554 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(140.),#830)
        REPRESENTATION_ITEM('maximum body height above seating plane'));
#555 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(62.),#830)
        REPRESENTATION_ITEM('maximum thickness'));
#556 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(62.),#830)
        REPRESENTATION_ITEM('minimum thickness'));
#557 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(150.),#830)
        REPRESENTATION_ITEM('minimum as finished passage extent'));
#558 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(61.),#830)
        REPRESENTATION_ITEM('minimum as finished passage extent'));
#559 = (LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(100.),#830)
        REPRESENTATION_ITEM('minimum as finished passage extent'));
#560 = PROPERTY_DEFINITION('located interconnect module thickness',
        '',#1415);
#561 = PROPERTY_DEFINITION('','',#486);
#562 = PROPERTY_DEFINITION('','',#487);
#563 = PROPERTY_DEFINITION('','',#488);

```

```
#564 = PROPERTY_DEFINITION(' ', ' ', #489);
#565 = PROPERTY_DEFINITION(' ', ' ', #490);
#566 = PROPERTY_DEFINITION(' ', ' ', #491);
#567 = PROPERTY_DEFINITION(' ', ' ', #492);
#568 = PROPERTY_DEFINITION(' ', ' ', #493);
#569 = PROPERTY_DEFINITION(' ', ' ', #494);
#570 = PROPERTY_DEFINITION(' ', ' ', #495);
#571 = PROPERTY_DEFINITION(' ', ' ', #496);
#572 = PROPERTY_DEFINITION(' ', ' ', #485);
#573 = PROPERTY_DEFINITION(' ', ' ', #419);
#574 = PROPERTY_DEFINITION(' ', ' ', #420);
#575 = PROPERTY_DEFINITION(' ', ' ', #421);
#576 = PROPERTY_DEFINITION(' ', ' ', #422);
#577 = PROPERTY_DEFINITION(' ', ' ', #423);
#578 = PROPERTY_DEFINITION(' ', ' ', #424);
#579 = PROPERTY_DEFINITION(' ', ' ', #425);
#580 = PROPERTY_DEFINITION(' ', ' ', #426);
#581 = PROPERTY_DEFINITION(' ', ' ', #427);
#582 = PROPERTY_DEFINITION(' ', ' ', #486);
#583 = PROPERTY_DEFINITION(' ', ' ', #487);
#584 = PROPERTY_DEFINITION(' ', ' ', #488);
#585 = PROPERTY_DEFINITION(' ', ' ', #489);
#586 = PROPERTY_DEFINITION(' ', ' ', #490);
#587 = PROPERTY_DEFINITION(' ', ' ', #491);
#588 = PROPERTY_DEFINITION(' ', ' ', #485);
#589 = PROPERTY_DEFINITION(' ', ' ', #492);
#590 = PROPERTY_DEFINITION(' ', ' ', #493);
#591 = PROPERTY_DEFINITION(' ', ' ', #494);
#592 = PROPERTY_DEFINITION(' ', ' ', #495);
#593 = PROPERTY_DEFINITION(' ', ' ', #496);
#594 = PROPERTY_DEFINITION(' ', ' ', #227);
#595 = PROPERTY_DEFINITION(' ', ' ', #204);
#596 = PROPERTY_DEFINITION(' ', ' ', #227);
#597 = PROPERTY_DEFINITION(' ', ' ', #227);
#598 = PROPERTY_DEFINITION(' ', ' ', #228);
#599 = PROPERTY_DEFINITION(' ', ' ', #205);
#600 = PROPERTY_DEFINITION(' ', ' ', #228);
#601 = PROPERTY_DEFINITION(' ', ' ', #228);
#602 = PROPERTY_DEFINITION(' ', ' ', #229);
#603 = PROPERTY_DEFINITION(' ', ' ', #206);
#604 = PROPERTY_DEFINITION(' ', ' ', #229);
#605 = PROPERTY_DEFINITION(' ', ' ', #229);
#606 = PROPERTY_DEFINITION(' ', ' ', #230);
#607 = PROPERTY_DEFINITION(' ', ' ', #207);
#608 = PROPERTY_DEFINITION(' ', ' ', #230);
#609 = PROPERTY_DEFINITION(' ', ' ', #230);
```

```
#610 = PROPERTY_DEFINITION(' ', ' ', #231);
#611 = PROPERTY_DEFINITION(' ', ' ', #208);
#612 = PROPERTY_DEFINITION(' ', ' ', #231);
#613 = PROPERTY_DEFINITION(' ', ' ', #231);
#614 = PROPERTY_DEFINITION(' ', ' ', #1414);
#615 = PROPERTY_DEFINITION(' ', ' ', #1414);
#616 = PROPERTY_DEFINITION(' ', ' ', #1001);
#617 = PROPERTY_DEFINITION(' ', ' ', #1002);
#618 = PROPERTY_DEFINITION(' ', ' ', #1003);
#619 = PROPERTY_DEFINITION(' ', ' ', #1004);
#620 = PROPERTY_DEFINITION(' ', ' ', #1005);
#621 = PROPERTY_DEFINITION(' ', ' ', #919);
#622 = PROPERTY_DEFINITION(' ', ' ', #920);
#623 = PROPERTY_DEFINITION(' ', ' ', #921);
#624 = PROPERTY_DEFINITION(' ', ' ', #922);
#625 = PROPERTY_DEFINITION(' ', ' ', #923);
#626 = PROPERTY_DEFINITION(' ', ' ', #919);
#627 = PROPERTY_DEFINITION(' ', ' ', #921);
#628 = PROPERTY_DEFINITION(' ', ' ', #922);
#629 = PROPERTY_DEFINITION(' ', ' ', #923);
#630 = PROPERTY_DEFINITION(' ', ' ', #887);
#631 = PROPERTY_DEFINITION(' ', ' ', #846);
#632 = PROPERTY_DEFINITION(' ', ' ', #888);
#633 = PROPERTY_DEFINITION(' ', ' ', #846);
#634 = PROPERTY_DEFINITION(' ', ' ', #889);
#635 = PROPERTY_DEFINITION(' ', ' ', #847);
#636 = PROPERTY_DEFINITION(' ', ' ', #847);
#637 = PROPERTY_DEFINITION(' ', ' ', #479);
#638 = PROPERTY_DEFINITION(' ', ' ', #480);
#639 = PROPERTY_DEFINITION(' ', ' ', #477);
#640 = PROPERTY_DEFINITION(' ', ' ', #478);
#641 = PROPERTY_DEFINITION(' ', ' ', #476);
#642 = PROPERTY_DEFINITION(' ', ' ', #481);
#643 = PROPERTY_DEFINITION(' ', ' ', #482);
#644 = PROPERTY_DEFINITION(' ', ' ', #483);
#645 = PROPERTY_DEFINITION(' ', ' ', #484);
#646 = PROPERTY_DEFINITION(' ', ' ', #1018);
#647 = PROPERTY_DEFINITION(' ', ' ', #1018);
#648 = PROPERTY_DEFINITION(' ', ' ', #1018);
#649 = PROPERTY_DEFINITION(' ', ' ', #1018);
#650 = PROPERTY_DEFINITION(' ', ' ', #1019);
#651 = PROPERTY_DEFINITION(' ', ' ', #1019);
#652 = PROPERTY_DEFINITION(' ', ' ', #1019);
#653 = PROPERTY_DEFINITION(' ', ' ', #1018);
#654 = PROPERTY_DEFINITION(' ', ' ', #1018);
#655 = PROPERTY_DEFINITION(' ', ' ', #1018);
```

```
#656 = PROPERTY_DEFINITION('', '', #1018);
#657 = PROPERTY_DEFINITION('', '', #1018);
#658 = PROPERTY_DEFINITION('', '', #1019);
#659 = PROPERTY_DEFINITION('', '', #1019);
#660 = PROPERTY_DEFINITION('', '', #1018);
#661 = PROPERTY_DEFINITION('', '', #1018);
#662 = PROPERTY_DEFINITION('', '', #1019);
#663 = PROPERTY_DEFINITION('', '', #1019);
#664 = PROPERTY_DEFINITION('', '', #1018);
#665 = PROPERTY_DEFINITION('', '', #1018);
#666 = PROPERTY_DEFINITION('', '', #1018);
#667 = PROPERTY_DEFINITION('', '', #1018);
#668 = PROPERTY_DEFINITION('', '', #1018);
#669 = PROPERTY_DEFINITION('', '', #1018);
#670 = PROPERTY_DEFINITION('', '', #1018);
#671 = PROPERTY_DEFINITION('', '', #227);
#672 = PROPERTY_DEFINITION('', '', #228);
#673 = PROPERTY_DEFINITION('', '', #229);
#674 = PROPERTY_DEFINITION('', '', #230);
#675 = PROPERTY_DEFINITION('', '', #231);
#676 = PROPERTY_DEFINITION('requirement', '', #227);
#677 = PROPERTY_DEFINITION('requirement', '', #228);
#678 = PROPERTY_DEFINITION('requirement', '', #229);
#679 = PROPERTY_DEFINITION('requirement', '', #230);
#680 = PROPERTY_DEFINITION('requirement', '', #231);
#681 = PROPERTY_DEFINITION_REPRESENTATION(#560, #1434);
#682 = PROPERTY_DEFINITION_REPRESENTATION(#1557, #1073);
#683 = PROPERTY_DEFINITION_REPRESENTATION(#1558, #1074);
#684 = PROPERTY_DEFINITION_REPRESENTATION(#1559, #1075);
#685 = PROPERTY_DEFINITION_REPRESENTATION(#1560, #1076);
#686 = PROPERTY_DEFINITION_REPRESENTATION(#1561, #1077);
#687 = PROPERTY_DEFINITION_REPRESENTATION(#1562, #1078);
#688 = PROPERTY_DEFINITION_REPRESENTATION(#1563, #1079);
#689 = PROPERTY_DEFINITION_REPRESENTATION(#1564, #1080);
#690 = PROPERTY_DEFINITION_REPRESENTATION(#1565, #1081);
#691 = PROPERTY_DEFINITION_REPRESENTATION(#561, #1439);
#692 = PROPERTY_DEFINITION_REPRESENTATION(#562, #1440);
#693 = PROPERTY_DEFINITION_REPRESENTATION(#563, #1441);
#694 = PROPERTY_DEFINITION_REPRESENTATION(#564, #1442);
#695 = PROPERTY_DEFINITION_REPRESENTATION(#565, #1443);
#696 = PROPERTY_DEFINITION_REPRESENTATION(#566, #1444);
#697 = PROPERTY_DEFINITION_REPRESENTATION(#567, #1445);
#698 = PROPERTY_DEFINITION_REPRESENTATION(#568, #1446);
#699 = PROPERTY_DEFINITION_REPRESENTATION(#569, #1447);
#700 = PROPERTY_DEFINITION_REPRESENTATION(#570, #1448);
#701 = PROPERTY_DEFINITION_REPRESENTATION(#571, #1449);
```


#702 = PROPERTY_DEFINITION_REPRESENTATION(#572,#1450);
#703 = PROPERTY_DEFINITION_REPRESENTATION(#573,#1452);
#704 = PROPERTY_DEFINITION_REPRESENTATION(#574,#1453);
#705 = PROPERTY_DEFINITION_REPRESENTATION(#575,#1454);
#706 = PROPERTY_DEFINITION_REPRESENTATION(#576,#1455);
#707 = PROPERTY_DEFINITION_REPRESENTATION(#577,#1456);
#708 = PROPERTY_DEFINITION_REPRESENTATION(#578,#1457);
#709 = PROPERTY_DEFINITION_REPRESENTATION(#579,#1458);
#710 = PROPERTY_DEFINITION_REPRESENTATION(#580,#1459);
#711 = PROPERTY_DEFINITION_REPRESENTATION(#581,#1460);
#712 = PROPERTY_DEFINITION_REPRESENTATION(#1596,#273);
#713 = PROPERTY_DEFINITION_REPRESENTATION(#1597,#278);
#714 = PROPERTY_DEFINITION_REPRESENTATION(#1598,#279);
#715 = PROPERTY_DEFINITION_REPRESENTATION(#1599,#280);
#716 = PROPERTY_DEFINITION_REPRESENTATION(#1600,#281);
#717 = PROPERTY_DEFINITION_REPRESENTATION(#1601,#282);
#718 = PROPERTY_DEFINITION_REPRESENTATION(#1602,#283);
#719 = PROPERTY_DEFINITION_REPRESENTATION(#1603,#284);
#720 = PROPERTY_DEFINITION_REPRESENTATION(#1604,#285);
#721 = PROPERTY_DEFINITION_REPRESENTATION(#1605,#286);
#722 = PROPERTY_DEFINITION_REPRESENTATION(#1606,#287);
#723 = PROPERTY_DEFINITION_REPRESENTATION(#1607,#288);
#724 = PROPERTY_DEFINITION_REPRESENTATION(#1608,#289);
#725 = PROPERTY_DEFINITION_REPRESENTATION(#1609,#290);
#726 = PROPERTY_DEFINITION_REPRESENTATION(#1610,#291);
#727 = PROPERTY_DEFINITION_REPRESENTATION(#1611,#292);
#728 = PROPERTY_DEFINITION_REPRESENTATION(#1612,#293);
#729 = PROPERTY_DEFINITION_REPRESENTATION(#1613,#294);
#730 = PROPERTY_DEFINITION_REPRESENTATION(#1614,#295);
#731 = PROPERTY_DEFINITION_REPRESENTATION(#1615,#296);
#732 = PROPERTY_DEFINITION_REPRESENTATION(#582,#1082);
#733 = PROPERTY_DEFINITION_REPRESENTATION(#1616,#271);
#734 = PROPERTY_DEFINITION_REPRESENTATION(#583,#1083);
#735 = PROPERTY_DEFINITION_REPRESENTATION(#1617,#272);
#736 = PROPERTY_DEFINITION_REPRESENTATION(#584,#1087);
#737 = PROPERTY_DEFINITION_REPRESENTATION(#1618,#274);
#738 = PROPERTY_DEFINITION_REPRESENTATION(#585,#1088);
#739 = PROPERTY_DEFINITION_REPRESENTATION(#1619,#275);
#740 = PROPERTY_DEFINITION_REPRESENTATION(#586,#1089);
#741 = PROPERTY_DEFINITION_REPRESENTATION(#1620,#276);
#742 = PROPERTY_DEFINITION_REPRESENTATION(#587,#1090);
#743 = PROPERTY_DEFINITION_REPRESENTATION(#1621,#277);
#744 = PROPERTY_DEFINITION_REPRESENTATION(#588,#1086);
#745 = PROPERTY_DEFINITION_REPRESENTATION(#1622,#297);
#746 = PROPERTY_DEFINITION_REPRESENTATION(#589,#1084);
#747 = PROPERTY_DEFINITION_REPRESENTATION(#1623,#298);

```
#748 = PROPERTY_DEFINITION_REPRESENTATION(#590,#1084);
#749 = PROPERTY_DEFINITION_REPRESENTATION(#1624,#299);
#750 = PROPERTY_DEFINITION_REPRESENTATION(#591,#1085);
#751 = PROPERTY_DEFINITION_REPRESENTATION(#1625,#300);
#752 = PROPERTY_DEFINITION_REPRESENTATION(#592,#1085);
#753 = PROPERTY_DEFINITION_REPRESENTATION(#1626,#301);
#754 = PROPERTY_DEFINITION_REPRESENTATION(#593,#1084);
#755 = PROPERTY_DEFINITION_REPRESENTATION(#1627,#302);
#756 = PROPERTY_DEFINITION_REPRESENTATION(#596,#1461);
#757 = PROPERTY_DEFINITION_REPRESENTATION(#597,#1462);
#758 = PROPERTY_DEFINITION_REPRESENTATION(#600,#1463);
#759 = PROPERTY_DEFINITION_REPRESENTATION(#601,#1464);
#760 = PROPERTY_DEFINITION_REPRESENTATION(#604,#1465);
#761 = PROPERTY_DEFINITION_REPRESENTATION(#605,#1466);
#762 = PROPERTY_DEFINITION_REPRESENTATION(#608,#1467);
#763 = PROPERTY_DEFINITION_REPRESENTATION(#609,#1468);
#764 = PROPERTY_DEFINITION_REPRESENTATION(#612,#1469);
#765 = PROPERTY_DEFINITION_REPRESENTATION(#613,#1470);
#766 = PROPERTY_DEFINITION_REPRESENTATION(#1006,#1424);
#767 = PROPERTY_DEFINITION_REPRESENTATION(#1007,#1425);
#768 = PROPERTY_DEFINITION_REPRESENTATION(#1007,#1426);
#769 = PROPERTY_DEFINITION_REPRESENTATION(#614,#1427);
#770 = PROPERTY_DEFINITION_REPRESENTATION(#615,#1428);
#771 = PROPERTY_DEFINITION_REPRESENTATION(#616,#1331);
#772 = PROPERTY_DEFINITION_REPRESENTATION(#617,#1332);
#773 = PROPERTY_DEFINITION_REPRESENTATION(#618,#1332);
#774 = PROPERTY_DEFINITION_REPRESENTATION(#619,#1333);
#775 = PROPERTY_DEFINITION_REPRESENTATION(#620,#1333);
#776 = PROPERTY_DEFINITION_REPRESENTATION(#621,#1429);
#777 = PROPERTY_DEFINITION_REPRESENTATION(#622,#1430);
#778 = PROPERTY_DEFINITION_REPRESENTATION(#623,#1431);
#779 = PROPERTY_DEFINITION_REPRESENTATION(#624,#1432);
#780 = PROPERTY_DEFINITION_REPRESENTATION(#625,#1433);
#781 = PROPERTY_DEFINITION_REPRESENTATION(#626,#1471);
#782 = PROPERTY_DEFINITION_REPRESENTATION(#627,#1472);
#783 = PROPERTY_DEFINITION_REPRESENTATION(#628,#1473);
#784 = PROPERTY_DEFINITION_REPRESENTATION(#629,#1474);
#785 = PROPERTY_DEFINITION_REPRESENTATION(#1638,#1337);
#786 = PROPERTY_DEFINITION_REPRESENTATION(#1639,#1072);
#787 = PROPERTY_DEFINITION_REPRESENTATION(#636,#1475);
#788 = PROPERTY_DEFINITION_REPRESENTATION(#637,#1082);
#789 = PROPERTY_DEFINITION_REPRESENTATION(#638,#1083);
#790 = PROPERTY_DEFINITION_REPRESENTATION(#639,#1084);
#791 = PROPERTY_DEFINITION_REPRESENTATION(#640,#1085);
#792 = PROPERTY_DEFINITION_REPRESENTATION(#641,#1086);
#793 = PROPERTY_DEFINITION_REPRESENTATION(#642,#1087);
```

```

#794 = PROPERTY_DEFINITION_REPRESENTATION(#643,#1088);
#795 = PROPERTY_DEFINITION_REPRESENTATION(#644,#1089);
#796 = PROPERTY_DEFINITION_REPRESENTATION(#645,#1090);
#797 = PROPERTY_DEFINITION_REPRESENTATION(#646,#278);
#798 = PROPERTY_DEFINITION_REPRESENTATION(#647,#279);
#799 = PROPERTY_DEFINITION_REPRESENTATION(#648,#280);
#800 = PROPERTY_DEFINITION_REPRESENTATION(#649,#281);
#801 = PROPERTY_DEFINITION_REPRESENTATION(#650,#282);
#802 = PROPERTY_DEFINITION_REPRESENTATION(#651,#283);
#803 = PROPERTY_DEFINITION_REPRESENTATION(#652,#284);
#804 = PROPERTY_DEFINITION_REPRESENTATION(#653,#285);
#805 = PROPERTY_DEFINITION_REPRESENTATION(#654,#286);
#806 = PROPERTY_DEFINITION_REPRESENTATION(#655,#287);
#807 = PROPERTY_DEFINITION_REPRESENTATION(#656,#288);
#808 = PROPERTY_DEFINITION_REPRESENTATION(#657,#289);
#809 = PROPERTY_DEFINITION_REPRESENTATION(#658,#290);
#810 = PROPERTY_DEFINITION_REPRESENTATION(#659,#291);
#811 = PROPERTY_DEFINITION_REPRESENTATION(#660,#292);
#812 = PROPERTY_DEFINITION_REPRESENTATION(#661,#293);
#813 = PROPERTY_DEFINITION_REPRESENTATION(#662,#294);
#814 = PROPERTY_DEFINITION_REPRESENTATION(#663,#295);
#815 = PROPERTY_DEFINITION_REPRESENTATION(#664,#296);
#816 = PROPERTY_DEFINITION_REPRESENTATION(#665,#297);
#817 = PROPERTY_DEFINITION_REPRESENTATION(#666,#298);
#818 = PROPERTY_DEFINITION_REPRESENTATION(#667,#299);
#819 = PROPERTY_DEFINITION_REPRESENTATION(#668,#300);
#820 = PROPERTY_DEFINITION_REPRESENTATION(#669,#301);
#821 = PROPERTY_DEFINITION_REPRESENTATION(#670,#302);
#822 = PROPERTY_DEFINITION_REPRESENTATION(#671,#1334);
#823 = PROPERTY_DEFINITION_REPRESENTATION(#672,#1334);
#824 = PROPERTY_DEFINITION_REPRESENTATION(#673,#1335);
#825 = PROPERTY_DEFINITION_REPRESENTATION(#674,#1335);
#826 = PROPERTY_DEFINITION_REPRESENTATION(#675,#1336);
#827 = ORGANIZATION('Intermedius Design Integration, LLC','','');
#828 = ORGANIZATION('PDES, INC','','');
#829 = (LENGTH_MEASURE_WITH_UNIT()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(5.E-06),#830)
      UNCERTAINTY_MEASURE_WITH_UNIT('',''));
#830 = (CONVERSION_BASED_UNIT('mil',#839)
      LENGTH_UNIT()
      NAMED_UNIT(#834));
#831 = (LENGTH_UNIT()
      NAMED_UNIT(*)
      SI_UNIT(.MILLI.,.METRE.));
#832 = PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(57.2958),#836);
#833 = DIMENSIONAL_EXPONENTS(0.,0.,0.,0.,0.,0.,0.);

```

```

#834 = DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
#835 = (CONVERSION_BASED_UNIT('degree',#832)
      NAMED_UNIT(#833)
      PLANE_ANGLE_UNIT());
#836 = (NAMED_UNIT(*)
      PLANE_ANGLE_UNIT()
      SI_UNIT($,.RADIAN.));
#837 = (GEOMETRIC_REPRESENTATION_CONTEXT(2)
      GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#829))
      GLOBAL_UNIT_ASSIGNED_CONTEXT((#835,#830))
      REPRESENTATION_CONTEXT('',''));
#838 = LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(100.),#830);
#839 = LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0254),#831);
#840 = STRUCTURED_TEXT_REPRESENTATION_CONTEXT('','');
#841 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #842,#843,#844,#845),
      #840,'text usage','');
#842 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #1666),#840,'','');
#843 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #1667),#840,'','');
#844 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #1668),#840,'','');
#845 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #1669),#840,'','');
#846 = STRATUM_TECHNOLOGY('conductive stuff','');
#847 = STRATUM_TECHNOLOGY('','');
#848 = MAPPED_ITEM(
      'planar positioned stratum shape in interconnect module',
      #270,#1476);
#849 = MAPPED_ITEM('component assembly 2d position',#232,#305);
#850 = MAPPED_ITEM('component assembly 2d position',#233,#310);
#851 = MAPPED_ITEM('component assembly 2d position',#234,#311);
#852 = MAPPED_ITEM('component assembly 2d position',#235,#312);
#853 = MAPPED_ITEM('component assembly 2d position',#236,#313);
#854 = MAPPED_ITEM('component assembly 2d position',#237,#314);
#855 = MAPPED_ITEM('component assembly 2d position',#238,#315);
#856 = MAPPED_ITEM('component assembly 2d position',#239,#316);
#857 = MAPPED_ITEM('component assembly 2d position',#240,#317);
#858 = MAPPED_ITEM('component assembly 2d position',#241,#318);
#859 = MAPPED_ITEM('component assembly 2d position',#242,#319);
#860 = MAPPED_ITEM('component assembly 2d position',#243,#320);
#861 = MAPPED_ITEM('component assembly 2d position',#244,#321);
#862 = MAPPED_ITEM('component assembly 2d position',#245,#322);
#863 = MAPPED_ITEM('component assembly 2d position',#246,#323);
#864 = MAPPED_ITEM('component assembly 2d position',#247,#324);
#865 = MAPPED_ITEM('component assembly 2d position',#248,#325);
#866 = MAPPED_ITEM('component assembly 2d position',#249,#326);
#867 = MAPPED_ITEM('component assembly 2d position',#250,#327);
#868 = MAPPED_ITEM('component assembly 2d position',#251,#328);
#869 = MAPPED_ITEM('component assembly 2d position',#252,#303);

```

```

#870 = MAPPED_ITEM('component assembly 2d position',#253,#304);
#871 = MAPPED_ITEM('component assembly 2d position',#254,#306);
#872 = MAPPED_ITEM('component assembly 2d position',#255,#307);
#873 = MAPPED_ITEM('component assembly 2d position',#256,#308);
#874 = MAPPED_ITEM('component assembly 2d position',#257,#309);
#875 = MAPPED_ITEM('component assembly 2d position',#258,#329);
#876 = MAPPED_ITEM('component assembly 2d position',#259,#330);
#877 = MAPPED_ITEM('component assembly 2d position',#260,#331);
#878 = MAPPED_ITEM('component assembly 2d position',#261,#332);
#879 = MAPPED_ITEM('component assembly 2d position',#262,#333);
#880 = MAPPED_ITEM('component assembly 2d position',#263,#334);
#881 = MAPPED_ITEM('',#264,#335);
#882 = MAPPED_ITEM('',#265,#336);
#883 = MAPPED_ITEM('',#266,#337);
#884 = MAPPED_ITEM('',#267,#338);
#885 = MAPPED_ITEM('',#268,#339);
#886 = MAPPED_ITEM('',#269,#340);
#887 = STRATUM('top','',#1371,#1727);
#888 = STRATUM('bottom','',#1371,#1727);
#889 = STRATUM('HEATSINK','',#1371,#1727);
#890 = PACKAGED_PART('890','packaged part',#1373,#1725);
#891 = PACKAGED_PART('891','packaged part',#1375,#1725);
#892 = PACKAGED_PART('892','packaged part',#1377,#1725);
#893 = PACKAGED_PART('893','packaged part',#1379,#1725);
#894 = PACKAGED_PART('894','packaged part',#1381,#1725);
#895 = PACKAGED_COMPONENT('895','',#1373,#1722);
#896 = PACKAGED_COMPONENT('896','',#1373,#1722);
#897 = PACKAGED_COMPONENT('897','',#1373,#1722);
#898 = PACKAGED_COMPONENT('898','',#1373,#1722);
#899 = PACKAGED_COMPONENT('899','',#1373,#1722);
#900 = PACKAGED_COMPONENT('900','',#1373,#1722);
#901 = PACKAGED_COMPONENT('901','',#1373,#1722);
#902 = PACKAGED_COMPONENT('902','',#1373,#1722);
#903 = PACKAGED_COMPONENT('903','',#1373,#1722);
#904 = PACKAGED_COMPONENT('904','',#1373,#1722);
#905 = PACKAGED_COMPONENT('905','',#1373,#1722);
#906 = PACKAGED_COMPONENT('906','',#1373,#1722);
#907 = PACKAGED_COMPONENT('907','',#1375,#1722);
#908 = PACKAGED_COMPONENT('908','',#1381,#1722);
#909 = PACKAGED_COMPONENT('909','',#1377,#1722);
#910 = PACKAGED_COMPONENT('910','',#1379,#1722);
#911 = PACKAGED_COMPONENT('911','',#1381,#1722);
#912 = PACKAGED_COMPONENT('912','',#1381,#1722);
#913 = PACKAGED_COMPONENT('913','',#1381,#1722);
#914 = PACKAGED_COMPONENT('914','',#1381,#1722);
#915 = PACKAGED_COMPONENT('915','',#1381,#1722);

```

```
#916 = PACKAGED_COMPONENT('916', '', #1381, #1722);
#917 = PACKAGED_COMPONENT('917', '', #1381, #1722);
#918 = PACKAGED_COMPONENT('918', '', #1377, #1722);
#919 = PACKAGE('919', 'package', #1374, #1725);
#920 = PACKAGE('920', 'package', #1376, #1725);
#921 = PACKAGE('921', 'package', #1378, #1725);
#922 = PACKAGE('922', 'package', #1380, #1725);
#923 = PACKAGE('923', 'package', #1382, #1725);
#924 = DIRECTION('', (1., 0.));
#925 = DIRECTION('', (1., 0.));
#926 = DIRECTION('', (1., 0.));
#927 = DIRECTION('', (1., 0.));
#928 = DIRECTION('', (1., 0.));
#929 = DIRECTION('', (1., 0.));
#930 = DIRECTION('', (1., 0.));
#931 = DIRECTION('', (1., 0.));
#932 = DIRECTION('', (1., 0.));
#933 = DIRECTION('', (1., 0.));
#934 = DIRECTION('', (1., 0.));
#935 = DIRECTION('', (1., 0.));
#936 = DIRECTION('', (-1., 0.));
#937 = DIRECTION('', (-1., 0.));
#938 = DIRECTION('', (-1., 0.));
#939 = DIRECTION('', (1., 0.));
#940 = DIRECTION('', (1., 0.));
#941 = DIRECTION('', (1., 0.));
#942 = DIRECTION('', (-1., 1.22464679914735E-16));
#943 = DIRECTION('', (1., 0.));
#944 = DIRECTION('', (-1., 0.));
#945 = DIRECTION('', (-1., 0.));
#946 = DIRECTION('', (1., 0.));
#947 = DIRECTION('', (1., 0.));
#948 = DIRECTION('', (-1., 0.));
#949 = DIRECTION('', (-1., 0.));
#950 = DIRECTION('', (1., 0.));
#951 = DIRECTION('', (1., 0.));
#952 = DIRECTION('', (1., 0.));
#953 = DIRECTION('', (1., 0.));
#954 = DIRECTION('', (1., 0.));
#955 = DIRECTION('', (1., 0.));
#956 = DIRECTION('', (1., 0.));
#957 = DIRECTION('', (1., 0.));
#958 = DIRECTION('', (0., 1.));
#959 = DIRECTION('', (1., 0.));
#960 = DIRECTION('', (0., 1.));
#961 = DIRECTION('', (1., 0.));
```

```

#962 = DIRECTION('', (0., 1.));
#963 = DIRECTION('', (1., 0.));
#964 = DIRECTION('', (0., 1.));
#965 = DIRECTION('', (1., 0.));
#966 = DIRECTION('', (0., 1.));
#967 = DIRECTION('', (1., 0.));
#968 = DIRECTION('', (0., 1.));
#969 = DIRECTION('', (0., 1.));
#970 = DIRECTION('', (0., 1.));
#971 = DIRECTION('', (0., 1.));
#972 = DIRECTION('', (0., 1.));
#973 = DIRECTION('', (0., 1.));
#974 = DIRECTION('', (0., 1.));
#975 = DIRECTION('', (0., 1.));
#976 = DIRECTION('', (0., 1.));
#977 = DIRECTION('', (0., 1.));
#978 = DIRECTION('', (0., 1.));
#979 = DIRECTION('', (0., 1.));
#980 = DIRECTION('', (0., 1.));
#981 = DIRECTION('', (0., 1.));
#982 = DIRECTION('', (0., 1.));
#983 = DIRECTION('', (0., 1.));
#984 = DIRECTION('', (0., 1.));
#985 = DIRECTION('', (0., 1.));
#986 = DIRECTION('', (0., 1.));
#987 = DIRECTION('', (0., 1.));
#988 = DIRECTION('', (0., 1.));
#989 = DIRECTION('', (0., 1.));
#990 = DIRECTION('', (-1.22464679914735E-16, -1.));
#991 = DIRECTION('', (0., 1.));
#992 = DIRECTION('', (0., 1.));
#993 = DIRECTION('', (0., 1.));
#994 = DIRECTION('', (0., 1.));
#995 = DIRECTION('', (0., 1.));
#996 = DIRECTION('', (0., 1.));
#997 = DIRECTION('', (0., 1.));
#998 = DIRECTION('', (0., 1.));
#999 = DIRECTION('', (0., 1.));
#1000 = DIRECTION('', (0., 1.));
#1001 = MOUNTING_RESTRICTION_AREA('', '', #1637, .F.);
#1002 = MOUNTING_RESTRICTION_AREA('', '', #1637, .F.);
#1003 = MOUNTING_RESTRICTION_AREA('', '', #1637, .F.);
#1004 = MOUNTING_RESTRICTION_AREA('', '', #1637, .F.);
#1005 = MOUNTING_RESTRICTION_AREA('', '', #1637, .F.);
#1006 = GROUPED_REQUIREMENTS_PROPERTY(
    'item restricted requirements property',

```

```

    '', 'rp 1', '', #1001);
#1007 = GROUPED_REQUIREMENTS_PROPERTY(
    'item restricted requirements property',
    '', 'rp 2', '', #1003);
#1008 = ASSEMBLY_COMPONENT_USAGE('',
    'interconnect module stratum assembly relationship',
    '', #1529, #887, $);
#1009 = ASSEMBLY_COMPONENT_USAGE('',
    'interconnect module stratum assembly relationship',
    '', #1529, #888, $);
#1010 = ASSEMBLY_COMPONENT_USAGE('',
    'interconnect module stratum assembly relationship',
    '', #1529, #889, 'other outline');
#1011 = SHAPE_ASPECT('1', 'interconnect module primary surface', #1559, .F.);
#1012 = SHAPE_ASPECT('2', 'interconnect module secondary surface', #1559, .F.);
#1013 = SHAPE_ASPECT('3', '', #1640, .F.);
#1014 = SHAPE_ASPECT('4', '', #1641, .F.);
#1015 = SHAPE_ASPECT('5', '', #1642, .F.);
#1016 = SHAPE_ASPECT('6', '', #1643, .F.);
#1017 = SHAPE_ASPECT('7', '', #1644, .F.);
#1018 = COMPONENT_SHAPE_ASPECT('',
    'interconnect module component surface feature', #1637, .F.);
#1019 = COMPONENT_SHAPE_ASPECT('',
    'interconnect module component surface feature', #1637, .F.);
#1020 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #887, #888);
#1021 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #887, #887);
#1022 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #888, #888);
#1023 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #889, #889);
#1024 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #889, #889);
#1025 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #889, #889);
#1026 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #889, #889);
#1027 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #1527, #1645);
#1028 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #1527, #1646);
#1029 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #1526, #1647);
#1030 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #890, #895);

```



```
#1031 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #896);  
#1032 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #897);  
#1033 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #898);  
#1034 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #899);  
#1035 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #900);  
#1036 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #901);  
#1037 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #902);  
#1038 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #903);  
#1039 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #904);  
#1040 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #905);  
#1041 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #890, #906);  
#1042 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #891, #907);  
#1043 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #908);  
#1044 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #892, #909);  
#1045 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #893, #910);  
#1046 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #911);  
#1047 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #912);  
#1048 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #913);  
#1049 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #914);  
#1050 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #915);  
#1051 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #916);  
#1052 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #894, #917);  
#1053 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',  
    '', #892, #918);
```

```

#1054 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package', '', #919, #890);
#1055 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package', '', #920, #891);
#1056 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package', '', #921, #892);
#1057 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package', '', #922, #893);
#1058 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package', '', #923, #894);
#1059 = PRODUCT_DEFINITION_RELATIONSHIP('', 'design usage', '', #1525, #1528);
#1060 = PRODUCT_DEFINITION_RELATIONSHIP('', 'design usage', '', #1526, #1529);
#1061 = PRODUCT_DEFINITION_RELATIONSHIP('', 'group member', '', #222, #1529);
#1062 = PRODUCT_DEFINITION_RELATIONSHIP('', 'group member', '', #223, #1529);
#1063 = PRODUCT_DEFINITION_RELATIONSHIP('', 'group member', '', #224, #1529);
#1064 = PRODUCT_DEFINITION_RELATIONSHIP('', 'group member', '', #225, #1529);
#1065 = PRODUCT_DEFINITION_RELATIONSHIP('', 'group member', '', #226, #1529);
#1066 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1483, #460), #837);
#1067 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1484, #456), #837);
#1068 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1485, #459), #837);
#1069 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1486, #457), #837);
#1070 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1487, #455), #837);
#1071 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION('',
    (#1488, #458), #837);
#1072 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1489, #450), #837);
#1073 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1490, #848, #869, #870, #871, #872, #873,
    #874, #875, #876, #877, #878, #879, #880), #837);
#1074 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1491, #849, #850, #851, #852, #853, #854,
    #855, #856, #857, #858, #859, #860, #861, #862, #863, #864, #865, #866,
    #867, #868), #837);
#1075 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1492, #448), #837);
#1076 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1493, #461), #837);
#1077 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1494, #462), #837);
#1078 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1495, #463), #837);
#1079 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1496, #464), #837);
#1080 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape', (#1497, #465), #837);
#1081 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(

```

```

        'planar projected shape', (#1498, #466), #837);
#1082 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1499, #448), #837);
#1083 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1500, #449), #837);
#1084 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1501, #445), #837);
#1085 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1503, #446), #837);
#1086 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1505, #447), #837);
#1087 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1507, #451), #837);
#1088 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1508, #452), #837);
#1089 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1509, #453), #837);
#1090 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
        'planar projected shape', (#1510, #454), #837);
#1091 = CARTESIAN_POINT('', (0., 0.));
#1092 = CARTESIAN_POINT('', (0., 0.));
#1093 = CARTESIAN_POINT('', (0., 0.));
#1094 = CARTESIAN_POINT('', (0., 0.));
#1095 = CARTESIAN_POINT('', (0., 0.));
#1096 = CARTESIAN_POINT('', (0., 0.));
#1097 = CARTESIAN_POINT('', (0., 0.));
#1098 = CARTESIAN_POINT('', (0., 0.));
#1099 = CARTESIAN_POINT('', (0., 0.));
#1100 = CARTESIAN_POINT('', (0., 0.));
#1101 = CARTESIAN_POINT('', (0., 0.));
#1102 = CARTESIAN_POINT('', (0., 0.));
#1103 = CARTESIAN_POINT('', (0., 0.));
#1104 = CARTESIAN_POINT('', (0., 0.));
#1105 = CARTESIAN_POINT('', (0., 0.));
#1106 = CARTESIAN_POINT('', (0., 0.));
#1107 = CARTESIAN_POINT('', (0., 0.));
#1108 = CARTESIAN_POINT('', (0., 0.));
#1109 = CARTESIAN_POINT('', (0., 0.));
#1110 = CARTESIAN_POINT('', (0., 0.));
#1111 = CARTESIAN_POINT('', (0., 0.));
#1112 = CARTESIAN_POINT('', (0., 0.));
#1113 = CARTESIAN_POINT('', (0., 0.));
#1114 = CARTESIAN_POINT('', (0., 0.));
#1115 = CARTESIAN_POINT('', (0., 0.));
#1116 = CARTESIAN_POINT('', (0., 0.));
#1117 = CARTESIAN_POINT('', (0., 0.));

```

```
#1118 = CARTESIAN_POINT('', (0., 0.));
#1119 = CARTESIAN_POINT('', (0., 0.));
#1120 = CARTESIAN_POINT('', (0., 0.));
#1121 = CARTESIAN_POINT('', (0., 0.));
#1122 = CARTESIAN_POINT('', (0., 0.));
#1123 = CARTESIAN_POINT('', (0., 0.));
#1124 = CARTESIAN_POINT('', (0., 0.));
#1125 = CARTESIAN_POINT('', (0., 0.));
#1126 = CARTESIAN_POINT('', (0., 0.));
#1127 = CARTESIAN_POINT('', (0., 0.));
#1128 = CARTESIAN_POINT('', (0., 0.));
#1129 = CARTESIAN_POINT('', (0., 0.));
#1130 = CARTESIAN_POINT('', (0., 0.));
#1131 = CARTESIAN_POINT('', (0., 0.));
#1132 = CARTESIAN_POINT('', (0., 0.));
#1133 = CARTESIAN_POINT('', (2150., 2425.));
#1134 = CARTESIAN_POINT('', (5825., 1800.));
#1135 = CARTESIAN_POINT('', (5825., 3050.));
#1136 = CARTESIAN_POINT('', (950., 1150.));
#1137 = CARTESIAN_POINT('', (2925., 1150.));
#1138 = CARTESIAN_POINT('', (1100., 1150.));
#1139 = CARTESIAN_POINT('', (3525., 1150.));
#1140 = CARTESIAN_POINT('', (2775., 2425.));
#1141 = CARTESIAN_POINT('', (4650., 1800.));
#1142 = CARTESIAN_POINT('', (2150., 1800.));
#1143 = CARTESIAN_POINT('', (5400., 300.));
#1144 = CARTESIAN_POINT('', (3100., 4450.));
#1145 = CARTESIAN_POINT('', (1925., 1000.));
#1146 = CARTESIAN_POINT('', (4400., 1000.));
#1147 = CARTESIAN_POINT('', (3875., 4450.));
#1148 = CARTESIAN_POINT('', (4675., 4450.));
#1149 = CARTESIAN_POINT('', (5625., 1000.));
#1150 = CARTESIAN_POINT('', (6630., 4750.));
#1151 = CARTESIAN_POINT('', (370., 4750.));
#1152 = CARTESIAN_POINT('', (6800., 4750.));
#1153 = CARTESIAN_POINT('', (1370., 200.));
#1154 = CARTESIAN_POINT('', (5650., 200.));
#1155 = CARTESIAN_POINT('', (6630., 4750.));
#1156 = CARTESIAN_POINT('', (370., 4750.));
#1157 = CARTESIAN_POINT('', (100., 300.));
#1158 = CARTESIAN_POINT('', (7000., 4750.));
#1159 = CARTESIAN_POINT('', (6750., 5000.));
#1160 = CARTESIAN_POINT('', (6750., 4750.));
#1161 = CARTESIAN_POINT('', (6750., 5000.));
#1162 = CARTESIAN_POINT('', (250., 5000.));
#1163 = CARTESIAN_POINT('', (0., 4750.));
```

```
#1164 = CARTESIAN_POINT('', (0., 250.));
#1165 = CARTESIAN_POINT('', (250., 0.));
#1166 = CARTESIAN_POINT('', (6750., 0.));
#1167 = CARTESIAN_POINT('', (7000., 250.));
#1168 = CARTESIAN_POINT('', (7000., 4750.));
#1169 = CARTESIAN_POINT('', (976.4, 4400.));
#1170 = CARTESIAN_POINT('', (1423.6, 4400.));
#1171 = CARTESIAN_POINT('', (1200., 4400.));
#1172 = CARTESIAN_POINT('', (1423.6, 4400.));
#1173 = CARTESIAN_POINT('', (976.4, 4400.));
#1174 = CARTESIAN_POINT('', (1200., 4400.));
#1175 = CARTESIAN_POINT('', (200., 250.));
#1176 = CARTESIAN_POINT('', (300., 150.));
#1177 = CARTESIAN_POINT('', (6700., 150.));
#1178 = CARTESIAN_POINT('', (6900., 350.));
#1179 = CARTESIAN_POINT('', (6900., 4650.));
#1180 = CARTESIAN_POINT('', (6700., 4850.));
#1181 = CARTESIAN_POINT('', (400., 4850.));
#1182 = CARTESIAN_POINT('', (200., 4650.));
#1183 = CARTESIAN_POINT('', (200., 250.));
#1184 = CARTESIAN_POINT('', (2800., 4450.));
#1185 = CARTESIAN_POINT('', (2850., 4500.));
#1186 = CARTESIAN_POINT('', (2850., 4450.));
#1187 = CARTESIAN_POINT('', (2850., 4500.));
#1188 = CARTESIAN_POINT('', (4250., 4500.));
#1189 = CARTESIAN_POINT('', (4250., 4500.));
#1190 = CARTESIAN_POINT('', (4300., 4450.));
#1191 = CARTESIAN_POINT('', (4250., 4450.));
#1192 = CARTESIAN_POINT('', (4300., 4450.));
#1193 = CARTESIAN_POINT('', (4300., 4350.));
#1194 = CARTESIAN_POINT('', (4300., 4350.));
#1195 = CARTESIAN_POINT('', (4250., 4300.));
#1196 = CARTESIAN_POINT('', (4250., 4350.));
#1197 = CARTESIAN_POINT('', (4250., 4300.));
#1198 = CARTESIAN_POINT('', (2850., 4300.));
#1199 = CARTESIAN_POINT('', (2850., 4300.));
#1200 = CARTESIAN_POINT('', (2800., 4350.));
#1201 = CARTESIAN_POINT('', (2850., 4350.));
#1202 = CARTESIAN_POINT('', (2800., 4350.));
#1203 = CARTESIAN_POINT('', (2800., 4450.));
#1204 = CARTESIAN_POINT('', (1440., 1400.));
#1205 = CARTESIAN_POINT('', (1320., 1280.));
#1206 = CARTESIAN_POINT('', (1320., 1400.));
#1207 = CARTESIAN_POINT('', (1320., 1280.));
#1208 = CARTESIAN_POINT('', (1270., 1280.));
#1209 = CARTESIAN_POINT('', (1270., 1280.));
```

```
#1210 = CARTESIAN_POINT('', (1150., 1400.));
#1211 = CARTESIAN_POINT('', (1270., 1400.));
#1212 = CARTESIAN_POINT('', (1150., 1400.));
#1213 = CARTESIAN_POINT('', (1150., 2021.6));
#1214 = CARTESIAN_POINT('', (1150., 2021.6));
#1215 = CARTESIAN_POINT('', (1440., 2016.45));
#1216 = CARTESIAN_POINT('', (1299.99100138312, 2300.07167982645));
#1217 = CARTESIAN_POINT('', (1440., 2016.45));
#1218 = CARTESIAN_POINT('', (1440., 2016.5));
#1219 = CARTESIAN_POINT('', (1440., 1400.));
#1220 = CARTESIAN_POINT('', (2500., 1800.));
#1221 = CARTESIAN_POINT('', (4200., 3500.));
#1222 = CARTESIAN_POINT('', (4400., 3300.));
#1223 = CARTESIAN_POINT('', (2700., 1600.));
#1224 = CARTESIAN_POINT('', (2500., 1800.));
#1225 = CARTESIAN_POINT('', (4952.8, 3600.));
#1226 = CARTESIAN_POINT('', (5847.2, 3600.));
#1227 = CARTESIAN_POINT('', (5400., 3600.));
#1228 = CARTESIAN_POINT('', (5847.2, 3600.));
#1229 = CARTESIAN_POINT('', (4952.8, 3600.));
#1230 = CARTESIAN_POINT('', (5400., 3600.));
#1231 = CARTESIAN_POINT('', (100., 400.));
#1232 = CARTESIAN_POINT('', (200., 300.));
#1233 = CARTESIAN_POINT('', (200., 200.));
#1234 = CARTESIAN_POINT('', (300., 100.));
#1235 = CARTESIAN_POINT('', (1200., 100.));
#1236 = CARTESIAN_POINT('', (1250., 50.));
#1237 = CARTESIAN_POINT('', (5850., 50.));
#1238 = CARTESIAN_POINT('', (5900., 100.));
#1239 = CARTESIAN_POINT('', (6700., 100.));
#1240 = CARTESIAN_POINT('', (6800., 200.));
#1241 = CARTESIAN_POINT('', (6800., 350.));
#1242 = CARTESIAN_POINT('', (6900., 450.));
#1243 = CARTESIAN_POINT('', (6900., 4650.));
#1244 = CARTESIAN_POINT('', (6600., 4650.));
#1245 = CARTESIAN_POINT('', (6350., 4900.));
#1246 = CARTESIAN_POINT('', (700., 4900.));
#1247 = CARTESIAN_POINT('', (650., 4900.));
#1248 = CARTESIAN_POINT('', (400., 4650.));
#1249 = CARTESIAN_POINT('', (100., 4650.));
#1250 = CARTESIAN_POINT('', (100., 2050.));
#1251 = CARTESIAN_POINT('', (100., 400.));
#1252 = CARTESIAN_POINT('', (250., 250.));
#1253 = CARTESIAN_POINT('', (6750., 250.));
#1254 = CARTESIAN_POINT('', (6750., 4440.));
#1255 = CARTESIAN_POINT('', (5750., 4890.));
```

```

#1256 = CARTESIAN_POINT('', (1250., 4890.));
#1257 = CARTESIAN_POINT('', (250., 4440.));
#1258 = CARTESIAN_POINT('', (250., 250.));
#1259 = CARTESIAN_POINT('', (600., 3400.));
#1260 = CARTESIAN_POINT('', (600., 2900.));
#1261 = CARTESIAN_POINT('', (1100., 2600.));
#1262 = CARTESIAN_POINT('', (1100., 3000.));
#1263 = CARTESIAN_POINT('', (1400., 3000.));
#1264 = CARTESIAN_POINT('', (1400., 3400.));
#1265 = CARTESIAN_POINT('', (600., 3400.));
#1266 = CARTESIAN_POINT('', (600., 3400.));
#1267 = CARTESIAN_POINT('', (600., 2900.));
#1268 = CARTESIAN_POINT('', (1100., 2600.));
#1269 = CARTESIAN_POINT('', (1100., 3000.));
#1270 = CARTESIAN_POINT('', (1400., 3000.));
#1271 = CARTESIAN_POINT('', (1400., 3400.));
#1272 = CARTESIAN_POINT('', (600., 3400.));
#1273 = CARTESIAN_POINT('', (300., 400.));
#1274 = CARTESIAN_POINT('', (6725., 400.));
#1275 = CARTESIAN_POINT('', (6725., 4875.));
#1276 = CARTESIAN_POINT('', (300., 4875.));
#1277 = CARTESIAN_POINT('', (300., 400.));
#1278 = CARTESIAN_POINT('', (300., 1100.));
#1279 = CARTESIAN_POINT('', (300., 400.));
#1280 = CARTESIAN_POINT('', (6700., 400.));
#1281 = CARTESIAN_POINT('', (6700., 1100.));
#1282 = CARTESIAN_POINT('', (300., 1100.));
#1283 = CARTESIAN_POINT('', (-49., 60.));
#1284 = CARTESIAN_POINT('', (-49., -60.));
#1285 = CARTESIAN_POINT('', (191., -60.));
#1286 = CARTESIAN_POINT('', (191., 60.));
#1287 = CARTESIAN_POINT('', (-49., 60.));
#1288 = CARTESIAN_POINT('', (-450., 300.));
#1289 = CARTESIAN_POINT('', (-450., -100.));
#1290 = CARTESIAN_POINT('', (4200., -100.));
#1291 = CARTESIAN_POINT('', (4200., 300.));
#1292 = CARTESIAN_POINT('', (-450., 300.));
#1293 = CARTESIAN_POINT('', (-172.5, 47.5));
#1294 = CARTESIAN_POINT('', (-220., -0.5));
#1295 = CARTESIAN_POINT('', (-220., -392.5));
#1296 = CARTESIAN_POINT('', (220., -392.5));
#1297 = CARTESIAN_POINT('', (220., 47.5));
#1298 = CARTESIAN_POINT('', (-172.5, 47.5));
#1299 = CARTESIAN_POINT('', (-172.5, 47.5));
#1300 = CARTESIAN_POINT('', (-220., -0.5));
#1301 = CARTESIAN_POINT('', (-220., -392.5));

```

```

#1302 = CARTESIAN_POINT('', (220., -392.5));
#1303 = CARTESIAN_POINT('', (220., 47.5));
#1304 = CARTESIAN_POINT('', (-172.5, 47.5));
#1305 = CARTESIAN_POINT('', (-232., 65.));
#1306 = CARTESIAN_POINT('', (-280., 17.));
#1307 = CARTESIAN_POINT('', (-280., -615.));
#1308 = CARTESIAN_POINT('', (280., -615.));
#1309 = CARTESIAN_POINT('', (280., 65.));
#1310 = CARTESIAN_POINT('', (-232., 65.));
#1311 = CARTESIAN_POINT('', (-200., 0.));
#1312 = CARTESIAN_POINT('', (-200., 300.));
#1313 = CARTESIAN_POINT('', (1000., 300.));
#1314 = CARTESIAN_POINT('', (1300., 500.));
#1315 = CARTESIAN_POINT('', (1300., 500.));
#1316 = CARTESIAN_POINT('', (1350., 450.));
#1317 = CARTESIAN_POINT('', (1350., 500.));
#1318 = CARTESIAN_POINT('', (1350., 450.));
#1319 = CARTESIAN_POINT('', (1300., 500.));
#1320 = CARTESIAN_POINT('', (800., 100.));
#1321 = CARTESIAN_POINT('', (0., 200.));
#1322 = CARTESIAN_POINT('', (0., 200.));
#1323 = CARTESIAN_POINT('', (-200., 0.));
#1324 = CARTESIAN_POINT('', (1.4210854715202E-14, -1.4210854715202E-14));
#1325 = CARTESIAN_POINT('', (0., 0.));
#1326 = CARTESIAN_POINT('', (0., 0.));
#1327 = CARTESIAN_POINT('', (0., 0.));
#1328 = CARTESIAN_POINT('', (0., 0.));
#1329 = CARTESIAN_POINT('', (0., 0.));
#1330 = CARTESIAN_POINT('', (0., 0.));
#1331 = SHAPE_REPRESENTATION('', (#1477, #881), #837);
#1332 = SHAPE_REPRESENTATION('', (#1478, #882), #837);
#1333 = SHAPE_REPRESENTATION('', (#1479, #883), #837);
#1334 = SHAPE_REPRESENTATION('', (#1480, #884), #837);
#1335 = SHAPE_REPRESENTATION('', (#1481, #885), #837);
#1336 = SHAPE_REPRESENTATION('', (#1482, #886), #837);
#1337 = SHAPE_REPRESENTATION('', (#848), #837);
#1338 = DOCUMENT_RELATIONSHIP('', '', #1344, #1340);
#1339 = DOCUMENT_RELATIONSHIP('', '', #1345, #1341);
#1340 = DOCUMENT('Preliminary', '', '', #1342);
#1341 = DOCUMENT('2.0', '', '', #1343);
#1342 = DOCUMENT_TYPE('design specification');
#1343 = DOCUMENT_TYPE('language reference manual');
#1344 = EE_SPECIFICATION('URLhttp://www.pdes.org',
    'EMpilot Working Paper', '', #1342);
#1345 = EE_SPECIFICATION('URLhttp://www.intermedius.com',
    'Intermediate Data Format Specification', '', #1343);

```



```

#1346 = PRODUCT('1', '', '', (#1733));
#1347 = PRODUCT('2', '', '', (#1733));
#1348 = PRODUCT('3', '', '', (#1733));
#1349 = PRODUCT('4', '', '', (#1733));
#1350 = PRODUCT('5', '', '', (#1733));
#1351 = PRODUCT('6', '', '', (#1733));
#1352 = PRODUCT('test_design_interconnect', '', '', (#1732));
#1353 = PRODUCT('test_design', '', '', (#1732));
#1354 = PRODUCT('pn-cap', '', '', (#1732));
#1355 = PRODUCT('pack-cc1210', '', '', (#1732));
#1356 = PRODUCT('pn-conn_152', '', '', (#1732));
#1357 = PRODUCT('pack-conn_152', '', '', (#1732));
#1358 = PRODUCT('IDT-54fct244LB.l', '', '', (#1732));
#1359 = PRODUCT('pack-lcc20 A', '', '', (#1732));
#1360 = PRODUCT('IDT-54fct245LB.l', '', '', (#1732));
#1361 = PRODUCT('pack-lcc20 B', '', '', (#1732));
#1362 = PRODUCT('IDT-71256s55LB.l', '', '', (#1732));
#1363 = PRODUCT('pack-lcc32', '', '', (#1732));
#1364 = PRODUCT('pn-extractor', '', '', (#1732));
#1365 = PRODUCT_DEFINITION_FORMATION('', '', #1346);
#1366 = PRODUCT_DEFINITION_FORMATION('', '', #1347);
#1367 = PRODUCT_DEFINITION_FORMATION('', '', #1348);
#1368 = PRODUCT_DEFINITION_FORMATION('', '', #1349);
#1369 = PRODUCT_DEFINITION_FORMATION('', '', #1350);
#1370 = PRODUCT_DEFINITION_FORMATION('', '', #1351);
#1371 = PRODUCT_DEFINITION_FORMATION('1', '', #1352);
#1372 = PRODUCT_DEFINITION_FORMATION('1', '', #1353);
#1373 = PRODUCT_DEFINITION_FORMATION('', '', #1354);
#1374 = PRODUCT_DEFINITION_FORMATION('', '', #1355);
#1375 = PRODUCT_DEFINITION_FORMATION('', '', #1356);
#1376 = PRODUCT_DEFINITION_FORMATION('', '', #1357);
#1377 = PRODUCT_DEFINITION_FORMATION('', '', #1358);
#1378 = PRODUCT_DEFINITION_FORMATION('', '', #1359);
#1379 = PRODUCT_DEFINITION_FORMATION('', '', #1360);
#1380 = PRODUCT_DEFINITION_FORMATION('', '', #1361);
#1381 = PRODUCT_DEFINITION_FORMATION('', '', #1362);
#1382 = PRODUCT_DEFINITION_FORMATION('', '', #1363);
#1383 = PRODUCT_DEFINITION_FORMATION('', '', #1364);
#1384 = PRODUCT_DEFINITION('1384', '', #1365, #1726);
#1385 = PRODUCT_DEFINITION('1385', '', #1366, #1726);
#1386 = PRODUCT_DEFINITION('1386', '', #1367, #1726);
#1387 = PRODUCT_DEFINITION('1387', '', #1368, #1726);
#1388 = PRODUCT_DEFINITION('1388', '', #1369, #1726);
#1389 = PRODUCT_DEFINITION('1389', '', #1370, #1726);
#1390 = PRODUCT_DEFINITION('1390', '', #1371, #1724);
#1391 = PRODUCT_DEFINITION('1391', '', #1371, #1724);

```

```

#1392 = PRODUCT_DEFINITION('1392', '', #1371, #1724);
#1393 = PRODUCT_DEFINITION('1393', '', #1371, #1724);
#1394 = PRODUCT_DEFINITION('1394', '', #1371, #1724);
#1395 = PRODUCT_DEFINITION('1395', '', #1371, #1724);
#1396 = PRODUCT_DEFINITION('1396', '', #1371, #1724);
#1397 = PRODUCT_DEFINITION('1397', '', #1371, #1724);
#1398 = PRODUCT_DEFINITION('1398', '', #1371, #1724);
#1399 = PRODUCT_DEFINITION('150 mil plated hole', 'passage technology',
    #1371, #1728);
#1400 = PRODUCT_DEFINITION('61 mil non-plated hole', 'passage technology',
    #1371, #1728);
#1401 = PRODUCT_DEFINITION('100 mil non-plated hole', 'passage technology',
    #1371, #1728);
#1402 = PRODUCT_DEFINITION('interconnect module edge', 'passage technology',
    #1371, #1728);
#1403 = PRODUCT_DEFINITION('cutout 1', 'passage technology', #1371, #1728);
#1404 = PRODUCT_DEFINITION('cutout 2', 'passage technology', #1371, #1728);
#1405 = PRODUCT_DEFINITION('cutout 3', 'passage technology', #1371, #1728);
#1406 = PRODUCT_DEFINITION('cutout 4', 'passage technology', #1371, #1728);
#1407 = PRODUCT_DEFINITION('cutout 5', 'passage technology', #1371, #1728);
#1408 = REQUIREMENTS_PROPERTY('1', 'design requirement', #1384);
#1409 = REQUIREMENTS_PROPERTY('2', 'design requirement', #1385);
#1410 = REQUIREMENTS_PROPERTY('3', 'design requirement', #1386);
#1411 = REQUIREMENTS_PROPERTY('4', 'design requirement', #1387);
#1412 = REQUIREMENTS_PROPERTY('5', 'design requirement', #1388);
#1413 = REQUIREMENTS_PROPERTY('6', 'design requirement', #1389);
#1414 = PHYSICAL_UNIT_DATUM('', 'plane', #155900, .F.);
#1415 = PHYSICAL_UNIT_DATUM('', 'point', #1559, .F.);
#1416 = COORDINATED_UNIVERSAL_TIME_OFFSET(0, 0, .BEHIND.);
#1417 = COORDINATED_UNIVERSAL_TIME_OFFSET(0, 0, .BEHIND.);
#1418 = LOCAL_TIME(0, 0, 0., #1416);
#1419 = LOCAL_TIME(14, 48, 3., #1417);
#1420 = CALENDAR_DATE(1993, 3, 1);
#1421 = CALENDAR_DATE(1992, 11, 11);
#1422 = DATE_AND_TIME(#1420, #1418);
#1423 = DATE_AND_TIME(#1421, #1419);
#1424 = REPRESENTATION('component placement restriction assignment',
    (#1663), #1729);
#1425 = REPRESENTATION('component placement restriction assignment',
    (#1664), #1729);
#1426 = REPRESENTATION('component placement restriction assignment',
    (#1665), #1729);
#1427 = REPRESENTATION('maximum positive component height', (#549), #1729);
#1428 = REPRESENTATION('minimum mounting clearance', (#550), #1729);
#1429 = REPRESENTATION('registered case style', (#1670), #1730);
#1430 = REPRESENTATION('registered case style', (#1671), #1730);

```

```

#1431 = REPRESENTATION('registered case style', (#1672), #1730);
#1432 = REPRESENTATION('registered case style', (#1673), #1730);
#1433 = REPRESENTATION('registered case style', (#1674), #1730);
#1434 = REPRESENTATION('', (#547, #548), #1729);
#1435 = REPRESENTATION('shape environment', (#1675), #1731);
#1436 = REPRESENTATION('shape material condition', (#1676), #1731);
#1437 = REPRESENTATION('shape purpose', (#1677), #1731);
#1438 = REPRESENTATION('shape purpose', (#1678), #1731);
#1439 = REPRESENTATION('', (#1662), #1729);
#1440 = REPRESENTATION('', (#1662), #1729);
#1441 = REPRESENTATION('', (#1662), #1729);
#1442 = REPRESENTATION('', (#1662), #1729);
#1443 = REPRESENTATION('', (#1662), #1729);
#1444 = REPRESENTATION('', (#1662), #1729);
#1445 = REPRESENTATION('', (#1662), #1729);
#1446 = REPRESENTATION('', (#1662), #1729);
#1447 = REPRESENTATION('', (#1662), #1729);
#1448 = REPRESENTATION('', (#1662), #1729);
#1449 = REPRESENTATION('', (#1662), #1729);
#1450 = REPRESENTATION('', (#1662), #1729);
#1451 = REPRESENTATION('shape environment', (#1679), #1731);
#1452 = REPRESENTATION('physical characteristics representation',
    (#1660, #557), #1729);
#1453 = REPRESENTATION('physical characteristics representation',
    (#1661, #558), #1729);
#1454 = REPRESENTATION('physical characteristics representation',
    (#1661, #559), #1729);
#1455 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1456 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1457 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1458 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1459 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1460 = REPRESENTATION('physical characteristics representation',
    (#1661), #1729);
#1461 = REPRESENTATION('', (#1712), #1729);
#1462 = REPRESENTATION('', (#1713), #1729);
#1463 = REPRESENTATION('', (#1714), #1729);
#1464 = REPRESENTATION('', (#1715), #1729);
#1465 = REPRESENTATION('', (#1716), #1729);
#1466 = REPRESENTATION('', (#1717), #1729);
#1467 = REPRESENTATION('', (#1718), #1729);

```

```

#1468 = REPRESENTATION('', (#1719), #1729);
#1469 = REPRESENTATION('', (#1720), #1729);
#1470 = REPRESENTATION('', (#1721), #1729);
#1471 = REPRESENTATION('package mounting data', (#551), #1729);
#1472 = REPRESENTATION('package mounting data', (#552), #1729);
#1473 = REPRESENTATION('package mounting data', (#553), #1729);
#1474 = REPRESENTATION('package mounting data', (#554), #1729);
#1475 = REPRESENTATION('physical characteristics representation',
    (#555, #556), #1729);
#1476 = AXIS2_PLACEMENT_2D('', #1128, #927);
#1477 = AXIS2_PLACEMENT_2D('origin', #1091, $);
#1478 = AXIS2_PLACEMENT_2D('origin', #1092, $);
#1479 = AXIS2_PLACEMENT_2D('origin', #1093, $);
#1480 = AXIS2_PLACEMENT_2D('origin', #1094, $);
#1481 = AXIS2_PLACEMENT_2D('origin', #1095, $);
#1482 = AXIS2_PLACEMENT_2D('origin', #1096, $);
#1483 = AXIS2_PLACEMENT_2D('origin', #1097, $);
#1484 = AXIS2_PLACEMENT_2D('origin', #1098, $);
#1485 = AXIS2_PLACEMENT_2D('origin', #1099, $);
#1486 = AXIS2_PLACEMENT_2D('origin', #1100, $);
#1487 = AXIS2_PLACEMENT_2D('origin', #1101, $);
#1488 = AXIS2_PLACEMENT_2D('origin', #1102, $);
#1489 = AXIS2_PLACEMENT_2D('origin', #1103, $);
#1490 = AXIS2_PLACEMENT_2D('origin', #1104, $);
#1491 = AXIS2_PLACEMENT_2D('origin', #1105, $);
#1492 = AXIS2_PLACEMENT_2D('origin', #1106, $);
#1493 = AXIS2_PLACEMENT_2D('origin', #1107, $);
#1494 = AXIS2_PLACEMENT_2D('origin', #1108, $);
#1495 = AXIS2_PLACEMENT_2D('origin', #1109, $);
#1496 = AXIS2_PLACEMENT_2D('origin', #1110, $);
#1497 = AXIS2_PLACEMENT_2D('origin', #1111, $);
#1498 = AXIS2_PLACEMENT_2D('origin', #1112, $);
#1499 = AXIS2_PLACEMENT_2D('origin', #1113, $);
#1500 = AXIS2_PLACEMENT_2D('origin', #1114, $);
#1501 = AXIS2_PLACEMENT_2D('origin', #1115, $);
#1502 = AXIS2_PLACEMENT_2D('', #1116, $);
#1503 = AXIS2_PLACEMENT_2D('origin', #1117, $);
#1504 = AXIS2_PLACEMENT_2D('', #1118, $);
#1505 = AXIS2_PLACEMENT_2D('origin', #1119, $);
#1506 = AXIS2_PLACEMENT_2D('', #1120, $);
#1507 = AXIS2_PLACEMENT_2D('origin', #1121, $);
#1508 = AXIS2_PLACEMENT_2D('origin', #1122, $);
#1509 = AXIS2_PLACEMENT_2D('origin', #1123, $);
#1510 = AXIS2_PLACEMENT_2D('origin', #1124, $);
#1511 = AXIS2_PLACEMENT_2D('', #1160, $);
#1512 = AXIS2_PLACEMENT_2D('', #1171, $);

```

```

#1513 = AXIS2_PLACEMENT_2D('', #1174, $);
#1514 = AXIS2_PLACEMENT_2D('', #1186, $);
#1515 = AXIS2_PLACEMENT_2D('', #1191, $);
#1516 = AXIS2_PLACEMENT_2D('', #1196, $);
#1517 = AXIS2_PLACEMENT_2D('', #1201, $);
#1518 = AXIS2_PLACEMENT_2D('', #1206, $);
#1519 = AXIS2_PLACEMENT_2D('', #1211, $);
#1520 = AXIS2_PLACEMENT_2D('', #1216, $);
#1521 = AXIS2_PLACEMENT_2D('', #1227, $);
#1522 = AXIS2_PLACEMENT_2D('', #1230, $);
#1523 = AXIS2_PLACEMENT_2D('', #1317, $);
#1524 = AXIS2_PLACEMENT_2D('', #1324, $);
#1525 = PHYSICAL_UNIT('1525', 'assembly module', #1372, #1725);
#1526 = PHYSICAL_UNIT('1526', 'interconnect module', #1371, #1725);
#1527 = PHYSICAL_UNIT('1527', '', #1383, #1725);
#1528 = PHYSICAL_UNIT('1528', 'assembly module', #1372, #1727);
#1529 = PHYSICAL_UNIT('1529', 'interconnect module', #1371, #1727);
#1530 = PRODUCT_DEFINITION_SHAPE('', '', #1645);
#1531 = PRODUCT_DEFINITION_SHAPE('', '', #1646);
#1533 = PRODUCT_DEFINITION_SHAPE('', '', #895);
#1534 = PRODUCT_DEFINITION_SHAPE('', '', #896);
#1535 = PRODUCT_DEFINITION_SHAPE('', '', #897);
#1536 = PRODUCT_DEFINITION_SHAPE('', '', #898);
#1537 = PRODUCT_DEFINITION_SHAPE('', '', #899);
#1538 = PRODUCT_DEFINITION_SHAPE('', '', #900);
#1539 = PRODUCT_DEFINITION_SHAPE('', '', #901);
#1540 = PRODUCT_DEFINITION_SHAPE('', '', #902);
#1541 = PRODUCT_DEFINITION_SHAPE('', '', #903);
#1542 = PRODUCT_DEFINITION_SHAPE('', '', #904);
#1543 = PRODUCT_DEFINITION_SHAPE('', '', #905);
#1544 = PRODUCT_DEFINITION_SHAPE('', '', #906);
#1545 = PRODUCT_DEFINITION_SHAPE('', '', #907);
#1546 = PRODUCT_DEFINITION_SHAPE('', '', #908);
#1547 = PRODUCT_DEFINITION_SHAPE('', '', #909);
#1548 = PRODUCT_DEFINITION_SHAPE('', '', #910);
#1549 = PRODUCT_DEFINITION_SHAPE('', '', #911);
#1550 = PRODUCT_DEFINITION_SHAPE('', '', #912);
#1551 = PRODUCT_DEFINITION_SHAPE('', '', #913);
#1552 = PRODUCT_DEFINITION_SHAPE('', '', #914);
#1553 = PRODUCT_DEFINITION_SHAPE('', '', #915);
#1554 = PRODUCT_DEFINITION_SHAPE('', '', #916);
#1555 = PRODUCT_DEFINITION_SHAPE('', '', #917);
#1556 = PRODUCT_DEFINITION_SHAPE('', '', #918);
#1557 = PRODUCT_DEFINITION_SHAPE('', '', #1529);
#1558 = PRODUCT_DEFINITION_SHAPE('', '', #1528);
#1559 = PRODUCT_DEFINITION_SHAPE('', '', #1526);

```

```
#155900 = PRODUCT_DEFINITION_SHAPE('', '#1525');
#1560 = PRODUCT_DEFINITION_SHAPE('', '#919');
#1561 = PRODUCT_DEFINITION_SHAPE('', '#920');
#1562 = PRODUCT_DEFINITION_SHAPE('', '#921');
#1563 = PRODUCT_DEFINITION_SHAPE('', '#922');
#1564 = PRODUCT_DEFINITION_SHAPE('', '#923');
#1565 = PRODUCT_DEFINITION_SHAPE('', '#1527');
#1566 = PRODUCT_DEFINITION_SHAPE('', '#1648');
#1567 = PRODUCT_DEFINITION_SHAPE('', '#1649');
#1568 = PRODUCT_DEFINITION_SHAPE('', '#1650');
#1569 = PRODUCT_DEFINITION_SHAPE('', '#1651');
#1570 = PRODUCT_DEFINITION_SHAPE('', '#1652');
#1571 = PRODUCT_DEFINITION_SHAPE('', '#1653');
#1572 = PRODUCT_DEFINITION_SHAPE('', '#1654');
#1573 = PRODUCT_DEFINITION_SHAPE('', '#1655');
#1574 = PRODUCT_DEFINITION_SHAPE('', '#1656');
#1575 = PRODUCT_DEFINITION_SHAPE('', '#1657');
#1576 = PRODUCT_DEFINITION_SHAPE('', '#1658');
#1577 = PRODUCT_DEFINITION_SHAPE('', '#1659');
#1578 = PRODUCT_DEFINITION_SHAPE('', '#1390');
#1579 = PRODUCT_DEFINITION_SHAPE('', '#1391');
#1580 = PRODUCT_DEFINITION_SHAPE('', '#1392');
#1581 = PRODUCT_DEFINITION_SHAPE('', '#1393');
#1582 = PRODUCT_DEFINITION_SHAPE('', '#1394');
#1583 = PRODUCT_DEFINITION_SHAPE('', '#1395');
#1584 = PRODUCT_DEFINITION_SHAPE('', '#1396');
#1585 = PRODUCT_DEFINITION_SHAPE('', '#1397');
#1586 = PRODUCT_DEFINITION_SHAPE('', '#1398');
#1587 = PRODUCT_DEFINITION_SHAPE('', '#1399');
#1588 = PRODUCT_DEFINITION_SHAPE('', '#1400');
#1589 = PRODUCT_DEFINITION_SHAPE('', '#1401');
#1590 = PRODUCT_DEFINITION_SHAPE('', '#1402');
#1591 = PRODUCT_DEFINITION_SHAPE('', '#1403');
#1592 = PRODUCT_DEFINITION_SHAPE('', '#1404');
#1593 = PRODUCT_DEFINITION_SHAPE('', '#1405');
#1594 = PRODUCT_DEFINITION_SHAPE('', '#1406');
#1595 = PRODUCT_DEFINITION_SHAPE('', '#1407');
#1596 = PRODUCT_DEFINITION_SHAPE('', '#382');
#1597 = PRODUCT_DEFINITION_SHAPE('', '#387');
#1598 = PRODUCT_DEFINITION_SHAPE('', '#388');
#1599 = PRODUCT_DEFINITION_SHAPE('', '#389');
#1600 = PRODUCT_DEFINITION_SHAPE('', '#390');
#1601 = PRODUCT_DEFINITION_SHAPE('', '#391');
#1602 = PRODUCT_DEFINITION_SHAPE('', '#392');
#1603 = PRODUCT_DEFINITION_SHAPE('', '#394');
#1604 = PRODUCT_DEFINITION_SHAPE('', '#396');
```

```

#1605 = PRODUCT_DEFINITION_SHAPE('', '#397');
#1606 = PRODUCT_DEFINITION_SHAPE('', '#398');
#1607 = PRODUCT_DEFINITION_SHAPE('', '#399');
#1608 = PRODUCT_DEFINITION_SHAPE('', '#400');
#1609 = PRODUCT_DEFINITION_SHAPE('', '#401');
#1610 = PRODUCT_DEFINITION_SHAPE('', '#402');
#1611 = PRODUCT_DEFINITION_SHAPE('', '#404');
#1612 = PRODUCT_DEFINITION_SHAPE('', '#405');
#1613 = PRODUCT_DEFINITION_SHAPE('', '#410');
#1614 = PRODUCT_DEFINITION_SHAPE('', '#411');
#1615 = PRODUCT_DEFINITION_SHAPE('', '#412');
#1616 = PRODUCT_DEFINITION_SHAPE('', '#380');
#1617 = PRODUCT_DEFINITION_SHAPE('', '#381');
#1618 = PRODUCT_DEFINITION_SHAPE('', '#383');
#1619 = PRODUCT_DEFINITION_SHAPE('', '#384');
#1620 = PRODUCT_DEFINITION_SHAPE('', '#385');
#1621 = PRODUCT_DEFINITION_SHAPE('', '#386');
#1622 = PRODUCT_DEFINITION_SHAPE('', '#413');
#1623 = PRODUCT_DEFINITION_SHAPE('', '#414');
#1624 = PRODUCT_DEFINITION_SHAPE('', '#415');
#1625 = PRODUCT_DEFINITION_SHAPE('', '#416');
#1626 = PRODUCT_DEFINITION_SHAPE('', '#417');
#1627 = PRODUCT_DEFINITION_SHAPE('', '#418');
#1628 = PRODUCT_DEFINITION_SHAPE('', '#222');
#1629 = PRODUCT_DEFINITION_SHAPE('', '#223');
#1630 = PRODUCT_DEFINITION_SHAPE('', '#224');
#1631 = PRODUCT_DEFINITION_SHAPE('', '#225');
#1632 = PRODUCT_DEFINITION_SHAPE('', '#226');
#1637 = PRODUCT_DEFINITION_SHAPE('', '#1647');
#1638 = PRODUCT_DEFINITION_SHAPE('', '#1010');
#1639 = PRODUCT_DEFINITION_SHAPE('', '#889');
#1640 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', '#1020');
#1641 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', '#1023');
#1642 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', '#1024');
#1643 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', '#1025');
#1644 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', '#1026');
#1645 = COMPONENT_DEFINITION('1645', '', '#1383, #1722');
#1646 = COMPONENT_DEFINITION('1646', '', '#1383, #1722');
#1647 = COMPONENT_DEFINITION('1647', 'interconnect module', '#1371, #1722');
#1648 = COMPONENT_DEFINITION('1648', 'interconnect module edge', '#1371, #1723');
#1649 = COMPONENT_DEFINITION('1649', 'cutout', '#1371, #1723');
#1650 = COMPONENT_DEFINITION('1650', 'cutout', '#1371, #1723');
#1651 = COMPONENT_DEFINITION('1651', 'cutout', '#1371, #1723');
#1652 = COMPONENT_DEFINITION('1652', 'cutout', '#1371, #1723');
#1653 = COMPONENT_DEFINITION('1653', 'cutout', '#1371, #1723');
#1654 = COMPONENT_DEFINITION('1654', 'non-plated thru-hole', '#1371, #1723);

```

```

#1655 = COMPONENT_DEFINITION('1655','non-plated thru-hole',#1371,#1723);
#1656 = COMPONENT_DEFINITION('1656','non-plated thru-hole',#1371,#1723);
#1657 = COMPONENT_DEFINITION('1657','non-plated thru-hole',#1371,#1723);
#1658 = COMPONENT_DEFINITION('1658','non-plated thru-hole',#1371,#1723);
#1659 = COMPONENT_DEFINITION('1659','plated thru-hole',#1371,#1723);
#1660 = DESCRIPTIVE_REPRESENTATION_ITEM('plated passage','true');
#1661 = DESCRIPTIVE_REPRESENTATION_ITEM('plated passage','false');
#1662 = DESCRIPTIVE_REPRESENTATION_ITEM('feature of size','false');
#1663 = DESCRIPTIVE_REPRESENTATION_ITEM('components permitted','false');
#1664 = DESCRIPTIVE_REPRESENTATION_ITEM('components permitted','true');
#1665 = DESCRIPTIVE_REPRESENTATION_ITEM('components permitted','true');
#1666 = DESCRIPTIVE_REPRESENTATION_ITEM('',
    'Design organization archival data set');
#1667 = DESCRIPTIVE_REPRESENTATION_ITEM('', 'Both the interconnect_module
    and the interconnect_module_component share the same
    geometric_curve_set');
#1668 = DESCRIPTIVE_REPRESENTATION_ITEM('', 'Added a representation_context
    and a representation that were not in the mapping tables, so the
    text_usage on an ee_document will pass WR1 for
    descriptive_representation_item');
#1669 = DESCRIPTIVE_REPRESENTATION_ITEM('', 'Added a representation_context
    with a context_type of "registered description" that was not in
    the mapping tables, so the context_of_items on a
    registered_case_style will pass the required attributes check');
#1670 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code','cc1210');
#1671 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code','conn_152');
#1672 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code','lcc20');
#1673 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code','lcc20');
#1674 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code','lcc32');
#1675 = DESCRIPTIVE_REPRESENTATION_ITEM('manufacturing','');
#1676 = DESCRIPTIVE_REPRESENTATION_ITEM('nominal material condition','');
#1677 = DESCRIPTIVE_REPRESENTATION_ITEM('design','');
#1678 = DESCRIPTIVE_REPRESENTATION_ITEM('design profile','');
#1679 = DESCRIPTIVE_REPRESENTATION_ITEM('end user application','');
#1680 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1681 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1682 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1683 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1684 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1685 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1686 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1687 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1688 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1689 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1690 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1691 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');

```



```

#1692 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1693 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1694 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1695 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1696 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1697 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1698 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1699 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1700 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1701 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1702 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1703 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1704 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1705 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#1706 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1707 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1708 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1709 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1710 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1711 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#1712 = DESCRIPTIVE_REPRESENTATION_ITEM('design specific purpose',
    'route keepout');
#1713 = DESCRIPTIVE_REPRESENTATION_ITEM('keepout','true');
#1714 = DESCRIPTIVE_REPRESENTATION_ITEM('design specific purpose',
    'route keepout');
#1715 = DESCRIPTIVE_REPRESENTATION_ITEM('keepout','true');
#1716 = DESCRIPTIVE_REPRESENTATION_ITEM('design specific purpose',
    'route outline');
#1717 = DESCRIPTIVE_REPRESENTATION_ITEM('keepout','false');
#1718 = DESCRIPTIVE_REPRESENTATION_ITEM('design specific purpose',
    'route outline');
#1719 = DESCRIPTIVE_REPRESENTATION_ITEM('keepout','false');
#1720 = DESCRIPTIVE_REPRESENTATION_ITEM('design specific purpose',
    'via keepout');
#1721 = DESCRIPTIVE_REPRESENTATION_ITEM('keepout','true');
#1722 = PRODUCT_DEFINITION_CONTEXT('physical occurrence',#1735,
    'as designed');
#1723 = PRODUCT_DEFINITION_CONTEXT('layout occurrence',#1735,
    'as designed');
#1724 = PRODUCT_DEFINITION_CONTEXT('template definition',#1735,
    'as designed');
#1725 = PRODUCT_DEFINITION_CONTEXT('physical design usage',#1735,
    'as designed');
#1726 = PRODUCT_DEFINITION_CONTEXT('design requirement',#1735,
    'as designed');
#1727 = PRODUCT_DEFINITION_CONTEXT('physical design',#1735,

```

```

        'as designed');
#1728 = PRODUCT_DEFINITION_CONTEXT('technology definition',#1735,
        'as designed');
#1729 = REPRESENTATION_CONTEXT('', '');
#1730 = REPRESENTATION_CONTEXT('', 'registered description');
#1731 = REPRESENTATION_CONTEXT('', 'shape conditions');
#1732 = PRODUCT_CONTEXT('', #1735, 'mechanical');
#1733 = PRODUCT_CONTEXT('requirements', #1735, '');
#1734 = APPLICATION_PROTOCOL_DEFINITION('',
        'electronic_assembly_interconnect_and_packaging_design',
        2000, #1735);
#1735 = APPLICATION_CONTEXT('EM pilot');
ENDSEC;
END-ISO-10303-21;

```

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(
/* description */ ('IDF/AP210 AIM file',
'EMPilot IDF to AP210 Translator Version 1.0'),
/* implementation_level */ '2;1');

```

```

FILE_NAME(
/* name */ 'SingleComponent210',
/* time_stamp */ '2000-08-21T13:29:06-07:00',
/* author */ ('Empilot Team'),
/* organization */ ('PDES Inc. EMPilot'),
/* preprocessor_version */ 'ST-DEVELOPER v7',
/* originating_system */ 'IDF Version 2.0',
/* authorisation */ '');

```

```

FILE_SCHEMA (('ELECTRONIC_ASSEMBLY_INTERCONNECT_AND_PACKAGING_DESIGN
{iso standard 10303 part(210) version(1) object(1)
electronic-assembly-interconnect-and-packaging-design(1)}'));
ENDSEC;

```

```

DATA;
#10 = PRODUCT_RELATED_PRODUCT_CATEGORY('assembly module', $, (#267));
#11 = PRODUCT_RELATED_PRODUCT_CATEGORY('interconnect module', $, (#266));
#12 = PROPERTY_DEFINITION_RELATIONSHIP('technology usage', '', #306, #305);
#13 = PROPERTY_DEFINITION_RELATIONSHIP('technology usage', '', #308, #307);
#14 = ORGANIZATION_ROLE('document source');
#15 = ORGANIZATION_ROLE('document source');
#16 = APPLIED_ORGANIZATION_ASSIGNMENT(#389, #14, (#535));
#17 = APPLIED_ORGANIZATION_ASSIGNMENT(#388, #15, (#536));

```

```

#18 = APPLIED_DOCUMENT_REFERENCE(#535, '', (#401));
#19 = APPLIED_DOCUMENT_REFERENCE(#536, '', (#433));
#20 = APPLIED_DOCUMENT_REFERENCE(#536, '', (#431));
#21 = APPLIED_DOCUMENT_REFERENCE(#536, '', (#547));
#22 = APPLIED_DOCUMENT_REFERENCE(#535, '', (#547));
#23 = APPLIED_DOCUMENT_REFERENCE(#536, '', (#548));
#24 = APPLIED_DOCUMENT_REFERENCE(#535, '', (#548));
#25 = DATE_TIME_ROLE('creation date');
#26 = DATE_TIME_ROLE('creation date');
#27 = DATE_TIME_ROLE('creation date');
#28 = DATE_TIME_ROLE('creation date');
#29 = DATE_TIME_ROLE('creation date');
#30 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#543, #25, (#536));
#31 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#544, #26, (#546));
#32 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#544, #27, (#545));
#33 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#544, #28, (#547));
#34 = APPLIED_DATE_AND_TIME_ASSIGNMENT(#544, #29, (#548));
#35 = SHAPE_ASPECT_RELATIONSHIP('instantiated feature', '', #464, #467);
#36 = SHAPE_ASPECT_RELATIONSHIP('instantiated feature', '', #465, #468);
#37 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #180);
#38 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #181);
#39 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #182);
#40 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #183);
#41 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #184);
#42 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #185);
#43 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #177, #186);
#44 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #178, #188);
#45 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #178, #189);
#46 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #178, #190);
#47 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #178, #191);
#48 = SHAPE_ASPECT_RELATIONSHIP('instantiated template', '', #179, #187);
#49 = SHAPE_ASPECT_RELATIONSHIP('via passage technology', '', #159, #177);
#50 = SHAPE_ASPECT_RELATIONSHIP('unsupported passage technology',
    '', #160, #178);
#51 = SHAPE_ASPECT_RELATIONSHIP('inter stratum feature passage technology',
    '', #161, #179);
#52 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #159, #466);
#53 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #160, #466);
#54 = SHAPE_ASPECT_RELATIONSHIP('as finished inter stratum extent',
    '', #161, #466);
#55 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #259, #259);
#56 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #257, #257);

```

```

#57 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#58 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#59 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#60 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#61 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#62 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#63 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #261, #261);
#64 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #260, #260);
#65 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #260, #260);
#66 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #260, #260);
#67 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #260, #260);
#68 = SHAPE_REPRESENTATION_RELATIONSHIP('component part planar shape',
    '', #258, #258);
#69 = POLYLINE('', (#236, #237, #238, #239, #240));
#70 = POLYLINE('', (#241, #242, #243, #244, #245));
#71 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #69);
#72 = COMPOSITE_CURVE_SEGMENT(.CONTINUOUS., .T., #70);
#73 = COMPOSITE_CURVE('', (#71), .T.);
#74 = COMPOSITE_CURVE('', (#72), .T.);
#75 = REPRESENTATION_MAP(#248, #257);
#76 = REPRESENTATION_MAP(#249, #258);
#77 = REPRESENTATION_MAP(#250, #259);
#78 = REPRESENTATION_MAP(#253, #261);
#79 = REPRESENTATION_MAP(#253, #261);
#80 = REPRESENTATION_MAP(#253, #261);
#81 = REPRESENTATION_MAP(#253, #261);
#82 = REPRESENTATION_MAP(#253, #261);
#83 = REPRESENTATION_MAP(#253, #261);
#84 = REPRESENTATION_MAP(#253, #261);
#85 = REPRESENTATION_MAP(#251, #260);
#86 = REPRESENTATION_MAP(#251, #260);
#87 = REPRESENTATION_MAP(#251, #260);
#88 = REPRESENTATION_MAP(#251, #260);
#89 = MAPPED_ITEM('component assembly 2d position', #75, #118);
#90 = MAPPED_ITEM('component assembly 2d position', #76, #119);

```

```

#91 = MAPPED_ITEM('component assembly 2d position',#77,#117);
#92 = MAPPED_ITEM('component assembly 2d position',#78,#120);
#93 = MAPPED_ITEM('component assembly 2d position',#79,#121);
#94 = MAPPED_ITEM('component assembly 2d position',#80,#122);
#95 = MAPPED_ITEM('component assembly 2d position',#81,#123);
#96 = MAPPED_ITEM('component assembly 2d position',#82,#124);
#97 = MAPPED_ITEM('component assembly 2d position',#83,#125);
#98 = MAPPED_ITEM('component assembly 2d position',#84,#126);
#99 = MAPPED_ITEM('component assembly 2d position',#85,#127);
#100 = MAPPED_ITEM('component assembly 2d position',#86,#128);
#101 = MAPPED_ITEM('component assembly 2d position',#87,#129);
#102 = MAPPED_ITEM('component assembly 2d position',#88,#130);
#103 = COMPONENT_LOCATION('planar projected shape',(#561,#117,#91),#398);
#104 = COMPONENT_LOCATION('planar projected shape',(#562,#118,#89),#398);
#105 = COMPONENT_LOCATION('planar projected shape',(#563,#119,#90),#398);
#106 = COMPONENT_LOCATION('planar projected shape',(#564,#120,#92),#398);
#107 = COMPONENT_LOCATION('planar projected shape',(#565,#121,#93),#398);
#108 = COMPONENT_LOCATION('planar projected shape',(#566,#122,#94),#398);
#109 = COMPONENT_LOCATION('planar projected shape',(#567,#123,#95),#398);
#110 = COMPONENT_LOCATION('planar projected shape',(#568,#124,#96),#398);
#111 = COMPONENT_LOCATION('planar projected shape',(#569,#125,#97),#398);
#112 = COMPONENT_LOCATION('planar projected shape',(#570,#126,#98),#398);
#113 = COMPONENT_LOCATION('planar projected shape',(#571,#127,#99),#398);
#114 = COMPONENT_LOCATION('planar projected shape',(#572,#128,#100),#398);
#115 = COMPONENT_LOCATION('planar projected shape',(#573,#129,#101),#398);
#116 = COMPONENT_LOCATION('planar projected shape',(#574,#130,#102),#398);
#117 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#434,#448,#222,1.);
#118 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#435,#449,#223,1.);
#119 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#436,#450,#224,1.);
#120 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#437,#451,#225,1.);
#121 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#438,#452,#226,1.);
#122 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#439,#453,#227,1.);
#123 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#440,#454,#228,1.);
#124 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#441,#455,#229,1.);
#125 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#442,#456,#230,1.);
#126 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#443,#457,#231,1.);
#127 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#444,#458,#232,1.);
#128 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#445,#459,#233,1.);
#129 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#446,#460,#234,1.);
#130 = CARTESIAN_TRANSFORMATION_OPERATOR_2D('',' ',#447,#461,#235,1.);
#131 = SHAPE_DEFINITION_REPRESENTATION(#475,#259);
#132 = SHAPE_DEFINITION_REPRESENTATION(#469,#257);
#133 = SHAPE_DEFINITION_REPRESENTATION(#480,#261);
#134 = SHAPE_DEFINITION_REPRESENTATION(#481,#261);
#135 = SHAPE_DEFINITION_REPRESENTATION(#482,#261);
#136 = SHAPE_DEFINITION_REPRESENTATION(#483,#261);

```

```

#137 = SHAPE_DEFINITION_REPRESENTATION(#484,#261);
#138 = SHAPE_DEFINITION_REPRESENTATION(#485,#261);
#139 = SHAPE_DEFINITION_REPRESENTATION(#486,#261);
#140 = SHAPE_DEFINITION_REPRESENTATION(#476,#260);
#141 = SHAPE_DEFINITION_REPRESENTATION(#477,#260);
#142 = SHAPE_DEFINITION_REPRESENTATION(#478,#260);
#143 = SHAPE_DEFINITION_REPRESENTATION(#479,#260);
#144 = SHAPE_DEFINITION_REPRESENTATION(#470,#258);
#145 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #509, $);
#146 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #547, #508, $);
#147 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #547, #432, 'U1');
#148 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #514, $);
#149 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #515, $);
#150 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #516, $);
#151 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #517, $);
#152 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #518, $);
#153 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #519, $);
#154 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #520, $);
#155 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #510, $);
#156 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #511, $);
#157 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #512, $);
#158 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('', 'assembly composition',
    '', #548, #513, $);
#159 = PASSAGE_TECHNOLOGY('62 mil plated hole', 'default via definition',
    #490, .F.);
#160 = PASSAGE_TECHNOLOGY('125 mil non-plated hole',
    'default unsupported passage definition', #491, .F.);
#161 = PASSAGE_TECHNOLOGY('interconnect module edge', '', #492, .F.);
#162 = CIRCLE('', #252, 62.5);
#163 = CIRCLE('', #254, 31.);
#164 = GEOMETRIC_CURVE_SET('', (#162));
#165 = GEOMETRIC_CURVE_SET('', (#163));
#166 = GEOMETRIC_CURVE_SET('', (#73));

```

```

#167 = GEOMETRIC_CURVE_SET('', (#74));
#168 = PRODUCT_DEFINITION('', '', #262, #577);
#169 = PRODUCT_DEFINITION('', '', #262, #577);
#170 = PRODUCT_DEFINITION('', '', #262, #577);
#171 = PRODUCT_DEFINITION('62 mil plated hole', 'passage technology',
    #262, #580);
#172 = PRODUCT_DEFINITION('125 mil non-plated hole', 'passage technology',
    #262, #580);
#173 = PRODUCT_DEFINITION('interconnect module edge', 'passage technology',
    #262, #580);
#174 = EXTERNAL_SOURCE(IDENTIFIER(''));
#175 = EXTERNAL_SOURCE(IDENTIFIER(''));
#176 = EXTERNAL_SOURCE(IDENTIFIER(''));
#177 = PART_TEMPLATE_DEFINITION('', 'via template',
    #487, .F., IDENTIFIER(''), #174);
#178 = PART_TEMPLATE_DEFINITION('', 'unsupported passage template',
    #488, .F., IDENTIFIER(''), #175);
#179 = PART_TEMPLATE_DEFINITION('', 'inter stratum feature template',
    #489, .F., IDENTIFIER(''), #176);
#180 = PLATED_PASSAGE('', 'interfacial connection', #480, .F.);
#181 = PLATED_PASSAGE('', 'interfacial connection', #481, .F.);
#182 = PLATED_PASSAGE('', 'interfacial connection', #482, .F.);
#183 = PLATED_PASSAGE('', 'interfacial connection', #483, .F.);
#184 = PLATED_PASSAGE('', 'interfacial connection', #484, .F.);
#185 = PLATED_PASSAGE('', 'interfacial connection', #485, .F.);
#186 = PLATED_PASSAGE('', 'interfacial connection', #486, .F.);
#187 = INTER_STRATUM_FEATURE('', 'interconnect module edge', #475, .F.);
#188 = INTER_STRATUM_FEATURE('', 'unsupported passage', #476, .F.);
#189 = INTER_STRATUM_FEATURE('', 'unsupported passage', #477, .F.);
#190 = INTER_STRATUM_FEATURE('', 'unsupported passage', #478, .F.);
#191 = INTER_STRATUM_FEATURE('', 'unsupported passage', #479, .F.);
#192 = REPRESENTATION_RELATIONSHIP('', '', #255, #411);
#193 = REPRESENTATION_RELATIONSHIP('', '', #255, #412);
#194 = REPRESENTATION_RELATIONSHIP('', '', #255, #413);
#195 = REPRESENTATION_RELATIONSHIP('', '', #256, #411);
#196 = REPRESENTATION_RELATIONSHIP('', '', #256, #412);
#197 = REPRESENTATION_RELATIONSHIP('', '', #256, #413);
#198 = REPRESENTATION_RELATIONSHIP('', '', #257, #411);
#199 = REPRESENTATION_RELATIONSHIP('', '', #257, #412);
#200 = REPRESENTATION_RELATIONSHIP('', '', #257, #413);
#201 = REPRESENTATION_RELATIONSHIP('', '', #258, #411);
#202 = REPRESENTATION_RELATIONSHIP('', '', #258, #412);
#203 = REPRESENTATION_RELATIONSHIP('', '', #258, #414);
#204 = REPRESENTATION_RELATIONSHIP('', '', #259, #411);
#205 = REPRESENTATION_RELATIONSHIP('', '', #259, #412);
#206 = REPRESENTATION_RELATIONSHIP('', '', #260, #411);

```

```

#207 = REPRESENTATION_RELATIONSHIP('', '#260, #412');
#208 = REPRESENTATION_RELATIONSHIP('', '#261, #411');
#209 = REPRESENTATION_RELATIONSHIP('', '#261, #412');
#210 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #402, #403);
#211 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #403, #404);
#212 = REPRESENTATION_RELATIONSHIP('precedent structured text',
    '', #404, #405);
#213 = CARTESIAN_POINT('', (0., 0.));
#214 = CARTESIAN_POINT('', (0., 0.));
#215 = CARTESIAN_POINT('', (0., 0.));
#216 = CARTESIAN_POINT('', (0., 0.));
#217 = CARTESIAN_POINT('', (0., 0.));
#218 = CARTESIAN_POINT('', (0., 0.));
#219 = CARTESIAN_POINT('', (0., 0.));
#220 = CARTESIAN_POINT('', (0., 0.));
#221 = CARTESIAN_POINT('', (0., 0.));
#222 = CARTESIAN_POINT('', (0., 0.));
#223 = CARTESIAN_POINT('', (0., 0.));
#224 = CARTESIAN_POINT('', (255., 1080.));
#225 = CARTESIAN_POINT('', (0., 1475.0004));
#226 = CARTESIAN_POINT('', (0., 1275.0005));
#227 = CARTESIAN_POINT('', (0., 1075.0004));
#228 = CARTESIAN_POINT('', (0., 875.0004));
#229 = CARTESIAN_POINT('', (0., 675.0004));
#230 = CARTESIAN_POINT('', (0., 475.0004));
#231 = CARTESIAN_POINT('', (0., 275.0004));
#232 = CARTESIAN_POINT('', (0., 0.0004));
#233 = CARTESIAN_POINT('', (500., 0.0004));
#234 = CARTESIAN_POINT('', (500., 1750.0004));
#235 = CARTESIAN_POINT('', (0., 1750.0004));
#236 = CARTESIAN_POINT('', (-125., 1875.0004));
#237 = CARTESIAN_POINT('', (-125., -124.9996));
#238 = CARTESIAN_POINT('', (625., -124.9996));
#239 = CARTESIAN_POINT('', (625., 1875.0004));
#240 = CARTESIAN_POINT('', (-125., 1875.0004));
#241 = CARTESIAN_POINT('', (-60., -60.));
#242 = CARTESIAN_POINT('', (-60., 60.));
#243 = CARTESIAN_POINT('', (60., 60.));
#244 = CARTESIAN_POINT('', (60., -60.));
#245 = CARTESIAN_POINT('', (-60., -60.));
#246 = AXIS2_PLACEMENT_2D('origin', #213, $);
#247 = AXIS2_PLACEMENT_2D('origin', #214, $);
#248 = AXIS2_PLACEMENT_2D('origin', #215, $);
#249 = AXIS2_PLACEMENT_2D('origin', #216, $);

```



```

#250 = AXIS2_PLACEMENT_2D('origin',#217,$);
#251 = AXIS2_PLACEMENT_2D('origin',#218,$);
#252 = AXIS2_PLACEMENT_2D('',#219,$);
#253 = AXIS2_PLACEMENT_2D('origin',#220,$);
#254 = AXIS2_PLACEMENT_2D('',#221,$);
#255 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',
    (#246,#91,#92,#93,#94,#95,#96,#97,#98,#99,#100,#101,#102),#398);
#256 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#247,#89,#90),#398);
#257 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#248,#166),#398);
#258 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#249,#167),#398);
#259 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#250,#166),#398);
#260 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#251,#164),#398);
#261 = GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION(
    'planar projected shape',(#253,#165),#398);
#262 = PRODUCT_DEFINITION_FORMATION('1','',#266);
#263 = PRODUCT_DEFINITION_FORMATION('1','',#267);
#264 = PRODUCT_DEFINITION_FORMATION('', '',#268);
#265 = PRODUCT_DEFINITION_FORMATION('', '',#269);
#266 = PRODUCT('design_interconnect','',',',(#584));
#267 = PRODUCT('design','',',',(#584));
#268 = PRODUCT('8FP_28_MANUAL_0.058','',',',(#584));
#269 = PRODUCT('pack-8FP_28_MANUAL_0.058','',',',(#584));
#270 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(60.),#391)
    REPRESENTATION_ITEM('maximum thickness'));
#271 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(60.),#391)
    REPRESENTATION_ITEM('minimum thickness'));
#272 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(58.),#391)
    REPRESENTATION_ITEM('maximum body height above seating plane'));
#273 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(62.),#391)
    REPRESENTATION_ITEM('minimum as finished passage extent'));
#274 = (LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()

```

```

        MEASURE_WITH_UNIT(LENGTH_MEASURE(125.),#391)
        REPRESENTATION_ITEM('minimum as finished passage extent'));
#275 = PROPERTY_DEFINITION('located interconnect module thickness',
    '',#387);
#276 = PROPERTY_DEFINITION('','',#187);
#277 = PROPERTY_DEFINITION('','',#188);
#278 = PROPERTY_DEFINITION('','',#189);
#279 = PROPERTY_DEFINITION('','',#190);
#280 = PROPERTY_DEFINITION('','',#191);
#281 = PROPERTY_DEFINITION('','',#180);
#282 = PROPERTY_DEFINITION('','',#181);
#283 = PROPERTY_DEFINITION('','',#182);
#284 = PROPERTY_DEFINITION('','',#183);
#285 = PROPERTY_DEFINITION('','',#184);
#286 = PROPERTY_DEFINITION('','',#185);
#287 = PROPERTY_DEFINITION('','',#186);
#288 = PROPERTY_DEFINITION('','',#159);
#289 = PROPERTY_DEFINITION('','',#160);
#290 = PROPERTY_DEFINITION('','',#161);
#291 = PROPERTY_DEFINITION('','',#187);
#292 = PROPERTY_DEFINITION('','',#180);
#293 = PROPERTY_DEFINITION('','',#181);
#294 = PROPERTY_DEFINITION('','',#182);
#295 = PROPERTY_DEFINITION('','',#183);
#296 = PROPERTY_DEFINITION('','',#184);
#297 = PROPERTY_DEFINITION('','',#185);
#298 = PROPERTY_DEFINITION('','',#186);
#299 = PROPERTY_DEFINITION('','',#188);
#300 = PROPERTY_DEFINITION('','',#189);
#301 = PROPERTY_DEFINITION('','',#190);
#302 = PROPERTY_DEFINITION('','',#191);
#303 = PROPERTY_DEFINITION('','',#433);
#304 = PROPERTY_DEFINITION('','',#433);
#305 = PROPERTY_DEFINITION('','',#407);
#306 = PROPERTY_DEFINITION('','',#406);
#307 = PROPERTY_DEFINITION('','',#408);
#308 = PROPERTY_DEFINITION('','',#406);
#309 = PROPERTY_DEFINITION('','',#179);
#310 = PROPERTY_DEFINITION('','',#178);
#311 = PROPERTY_DEFINITION('','',#177);
#312 = PROPERTY_DEFINITION('','',#467);
#313 = PROPERTY_DEFINITION('','',#467);
#314 = PROPERTY_DEFINITION('','',#467);
#315 = PROPERTY_DEFINITION('','',#467);
#316 = PROPERTY_DEFINITION('','',#467);
#317 = PROPERTY_DEFINITION('','',#467);

```

```
#318 = PROPERTY_DEFINITION(' ', ' ', #467);
#319 = PROPERTY_DEFINITION(' ', ' ', #467);
#320 = PROPERTY_DEFINITION(' ', ' ', #467);
#321 = PROPERTY_DEFINITION(' ', ' ', #467);
#322 = PROPERTY_DEFINITION(' ', ' ', #467);
#323 = PROPERTY_DEFINITION(' ', ' ', #467);
#324 = PROPERTY_DEFINITION_REPRESENTATION(#275, #410);
#325 = PROPERTY_DEFINITION_REPRESENTATION(#471, #255);
#326 = PROPERTY_DEFINITION_REPRESENTATION(#472, #256);
#327 = PROPERTY_DEFINITION_REPRESENTATION(#473, #257);
#328 = PROPERTY_DEFINITION_REPRESENTATION(#474, #258);
#329 = PROPERTY_DEFINITION_REPRESENTATION(#276, #415);
#330 = PROPERTY_DEFINITION_REPRESENTATION(#277, #416);
#331 = PROPERTY_DEFINITION_REPRESENTATION(#278, #417);
#332 = PROPERTY_DEFINITION_REPRESENTATION(#279, #418);
#333 = PROPERTY_DEFINITION_REPRESENTATION(#280, #419);
#334 = PROPERTY_DEFINITION_REPRESENTATION(#281, #420);
#335 = PROPERTY_DEFINITION_REPRESENTATION(#282, #421);
#336 = PROPERTY_DEFINITION_REPRESENTATION(#283, #422);
#337 = PROPERTY_DEFINITION_REPRESENTATION(#284, #423);
#338 = PROPERTY_DEFINITION_REPRESENTATION(#285, #424);
#339 = PROPERTY_DEFINITION_REPRESENTATION(#286, #425);
#340 = PROPERTY_DEFINITION_REPRESENTATION(#287, #426);
#341 = PROPERTY_DEFINITION_REPRESENTATION(#288, #427);
#342 = PROPERTY_DEFINITION_REPRESENTATION(#289, #428);
#343 = PROPERTY_DEFINITION_REPRESENTATION(#290, #429);
#344 = PROPERTY_DEFINITION_REPRESENTATION(#493, #104);
#345 = PROPERTY_DEFINITION_REPRESENTATION(#494, #105);
#346 = PROPERTY_DEFINITION_REPRESENTATION(#291, #259);
#347 = PROPERTY_DEFINITION_REPRESENTATION(#495, #103);
#348 = PROPERTY_DEFINITION_REPRESENTATION(#292, #261);
#349 = PROPERTY_DEFINITION_REPRESENTATION(#496, #106);
#350 = PROPERTY_DEFINITION_REPRESENTATION(#293, #261);
#351 = PROPERTY_DEFINITION_REPRESENTATION(#497, #107);
#352 = PROPERTY_DEFINITION_REPRESENTATION(#294, #261);
#353 = PROPERTY_DEFINITION_REPRESENTATION(#498, #108);
#354 = PROPERTY_DEFINITION_REPRESENTATION(#295, #261);
#355 = PROPERTY_DEFINITION_REPRESENTATION(#499, #109);
#356 = PROPERTY_DEFINITION_REPRESENTATION(#296, #261);
#357 = PROPERTY_DEFINITION_REPRESENTATION(#500, #110);
#358 = PROPERTY_DEFINITION_REPRESENTATION(#297, #261);
#359 = PROPERTY_DEFINITION_REPRESENTATION(#501, #111);
#360 = PROPERTY_DEFINITION_REPRESENTATION(#298, #261);
#361 = PROPERTY_DEFINITION_REPRESENTATION(#502, #112);
#362 = PROPERTY_DEFINITION_REPRESENTATION(#299, #260);
#363 = PROPERTY_DEFINITION_REPRESENTATION(#503, #113);
```

```

#364 = PROPERTY_DEFINITION_REPRESENTATION(#300,#260);
#365 = PROPERTY_DEFINITION_REPRESENTATION(#504,#114);
#366 = PROPERTY_DEFINITION_REPRESENTATION(#301,#260);
#367 = PROPERTY_DEFINITION_REPRESENTATION(#505,#115);
#368 = PROPERTY_DEFINITION_REPRESENTATION(#302,#260);
#369 = PROPERTY_DEFINITION_REPRESENTATION(#506,#116);
#370 = PROPERTY_DEFINITION_REPRESENTATION(#303,#409);
#371 = PROPERTY_DEFINITION_REPRESENTATION(#304,#430);
#372 = PROPERTY_DEFINITION_REPRESENTATION(#309,#259);
#373 = PROPERTY_DEFINITION_REPRESENTATION(#310,#260);
#374 = PROPERTY_DEFINITION_REPRESENTATION(#311,#261);
#375 = PROPERTY_DEFINITION_REPRESENTATION(#312,#105);
#376 = PROPERTY_DEFINITION_REPRESENTATION(#313,#106);
#377 = PROPERTY_DEFINITION_REPRESENTATION(#314,#107);
#378 = PROPERTY_DEFINITION_REPRESENTATION(#315,#108);
#379 = PROPERTY_DEFINITION_REPRESENTATION(#316,#109);
#380 = PROPERTY_DEFINITION_REPRESENTATION(#317,#110);
#381 = PROPERTY_DEFINITION_REPRESENTATION(#318,#111);
#382 = PROPERTY_DEFINITION_REPRESENTATION(#319,#112);
#383 = PROPERTY_DEFINITION_REPRESENTATION(#320,#113);
#384 = PROPERTY_DEFINITION_REPRESENTATION(#321,#114);
#385 = PROPERTY_DEFINITION_REPRESENTATION(#322,#115);
#386 = PROPERTY_DEFINITION_REPRESENTATION(#323,#116);
#387 = PHYSICAL_UNIT_DATUM('', 'point', #473, .F.);
#388 = ORGANIZATION('Intermedius Design Integration, LLC', '', '');
#389 = ORGANIZATION('PDES, INC', '', '');
#390 = (LENGTH_MEASURE_WITH_UNIT(
    MEASURE_WITH_UNIT(LENGTH_MEASURE(5.E-006), #391)
    UNCERTAINTY_MEASURE_WITH_UNIT('', ''));
#391 = (CONVERSION_BASED_UNIT('mil', #399)
    LENGTH_UNIT(
    NAMED_UNIT(#395));
#392 = (LENGTH_UNIT(
    NAMED_UNIT(*)
    SI_UNIT(.MILLI., .METRE.));
#393 = PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(57.2958), #397);
#394 = DIMENSIONAL_EXPONENTS(0., 0., 0., 0., 0., 0., 0.);
#395 = DIMENSIONAL_EXPONENTS(1., 0., 0., 0., 0., 0., 0.);
#396 = (CONVERSION_BASED_UNIT('degree', #393)
    NAMED_UNIT(#394)
    PLANE_ANGLE_UNIT());
#397 = (NAMED_UNIT(*)
    PLANE_ANGLE_UNIT(
    SI_UNIT($, .RADIAN.));
#398 = (GEOMETRIC_REPRESENTATION_CONTEXT(2)
    GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#390))

```

```

GLOBAL_UNIT_ASSIGNED_CONTEXT((#396,#391))
REPRESENTATION_CONTEXT('','');
#399 = LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0254),#392);
#400 = STRUCTURED_TEXT_REPRESENTATION_CONTEXT('','');
#401 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #402,#403,#404,#405),
#400,'text usage','');
#402 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #552),#400,'','');
#403 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #553),#400,'','');
#404 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #554),#400,'','');
#405 = STRUCTURED_TEXT_REPRESENTATION_ITEM('',( #555),#400,'','');
#406 = STRATUM_TECHNOLOGY('conductive stuff','');
#407 = STRATUM('top','','#262,#579);
#408 = STRATUM('bottom','','#262,#579);
#409 = REPRESENTATION('registered case style',( #556),#582);
#410 = REPRESENTATION('',( #270,#271),#581);
#411 = REPRESENTATION('shape environment',( #557),#583);
#412 = REPRESENTATION('shape material condition',( #558),#583);
#413 = REPRESENTATION('shape purpose',( #559),#583);
#414 = REPRESENTATION('shape purpose',( #560),#583);
#415 = REPRESENTATION('',( #551),#581);
#416 = REPRESENTATION('',( #551),#581);
#417 = REPRESENTATION('',( #551),#581);
#418 = REPRESENTATION('',( #551),#581);
#419 = REPRESENTATION('',( #551),#581);
#420 = REPRESENTATION('',( #551),#581);
#421 = REPRESENTATION('',( #551),#581);
#422 = REPRESENTATION('',( #551),#581);
#423 = REPRESENTATION('',( #551),#581);
#424 = REPRESENTATION('',( #551),#581);
#425 = REPRESENTATION('',( #551),#581);
#426 = REPRESENTATION('',( #551),#581);
#427 = REPRESENTATION('physical characteristics representation',
( #549,#273),#581);
#428 = REPRESENTATION('physical characteristics representation',
( #550,#274),#581);
#429 = REPRESENTATION('physical characteristics representation',
( #550),#581);
#430 = REPRESENTATION('package mounting data',( #272),#581);
#431 = PACKAGED_PART('','packaged part',#264,#578);
#432 = PACKAGED_COMPONENT('','',#264,#575);
#433 = PACKAGE('','package',#265,#578);
#434 = DIRECTION('',(1.,0.));
#435 = DIRECTION('',(1.,0.));
#436 = DIRECTION('',(-1.,1.22460635382238E-016));
#437 = DIRECTION('',(1.,0.));
#438 = DIRECTION('',(1.,0.));

```

```

#439 = DIRECTION('', (1., 0.));
#440 = DIRECTION('', (1., 0.));
#441 = DIRECTION('', (1., 0.));
#442 = DIRECTION('', (1., 0.));
#443 = DIRECTION('', (1., 0.));
#444 = DIRECTION('', (1., 0.));
#445 = DIRECTION('', (1., 0.));
#446 = DIRECTION('', (1., 0.));
#447 = DIRECTION('', (1., 0.));
#448 = DIRECTION('', (0., 1.));
#449 = DIRECTION('', (0., 1.));
#450 = DIRECTION('', (-1.22460635382238E-016, -1.));
#451 = DIRECTION('', (0., 1.));
#452 = DIRECTION('', (0., 1.));
#453 = DIRECTION('', (0., 1.));
#454 = DIRECTION('', (0., 1.));
#455 = DIRECTION('', (0., 1.));
#456 = DIRECTION('', (0., 1.));
#457 = DIRECTION('', (0., 1.));
#458 = DIRECTION('', (0., 1.));
#459 = DIRECTION('', (0., 1.));
#460 = DIRECTION('', (0., 1.));
#461 = DIRECTION('', (0., 1.));
#462 = ASSEMBLY_COMPONENT_USAGE('',
    'interconnect module stratum assembly relationship', '', #548, #407, $);
#463 = ASSEMBLY_COMPONENT_USAGE('',
    'interconnect module stratum assembly relationship', '', #548, #408, $);
#464 = SHAPE_ASPECT('', 'interconnect module primary surface', #473, .F.);
#465 = SHAPE_ASPECT('', 'interconnect module secondary surface', #473, .F.);
#466 = SHAPE_ASPECT('', '', #507, .F.);
#467 = COMPONENT_SHAPE_ASPECT('',
    'interconnect module component surface feature', #469, .F.);
#468 = COMPONENT_SHAPE_ASPECT('',
    'interconnect module component surface feature', #469, .F.);
#469 = PRODUCT_DEFINITION_SHAPE('', '', #508);
#470 = PRODUCT_DEFINITION_SHAPE('', '', #432);
#471 = PRODUCT_DEFINITION_SHAPE('', '', #548);
#472 = PRODUCT_DEFINITION_SHAPE('', '', #547);
#473 = PRODUCT_DEFINITION_SHAPE('', '', #546);
#474 = PRODUCT_DEFINITION_SHAPE('', '', #433);
#475 = PRODUCT_DEFINITION_SHAPE('', '', #509);
#476 = PRODUCT_DEFINITION_SHAPE('', '', #510);
#477 = PRODUCT_DEFINITION_SHAPE('', '', #511);
#478 = PRODUCT_DEFINITION_SHAPE('', '', #512);
#479 = PRODUCT_DEFINITION_SHAPE('', '', #513);
#480 = PRODUCT_DEFINITION_SHAPE('', '', #514);

```

```

#481 = PRODUCT_DEFINITION_SHAPE('', '#515);
#482 = PRODUCT_DEFINITION_SHAPE('', '#516);
#483 = PRODUCT_DEFINITION_SHAPE('', '#517);
#484 = PRODUCT_DEFINITION_SHAPE('', '#518);
#485 = PRODUCT_DEFINITION_SHAPE('', '#519);
#486 = PRODUCT_DEFINITION_SHAPE('', '#520);
#487 = PRODUCT_DEFINITION_SHAPE('', '#168);
#488 = PRODUCT_DEFINITION_SHAPE('', '#169);
#489 = PRODUCT_DEFINITION_SHAPE('', '#170);
#490 = PRODUCT_DEFINITION_SHAPE('', '#171);
#491 = PRODUCT_DEFINITION_SHAPE('', '#172);
#492 = PRODUCT_DEFINITION_SHAPE('', '#173);
#493 = PRODUCT_DEFINITION_SHAPE('', '#146);
#494 = PRODUCT_DEFINITION_SHAPE('', '#147);
#495 = PRODUCT_DEFINITION_SHAPE('', '#145);
#496 = PRODUCT_DEFINITION_SHAPE('', '#148);
#497 = PRODUCT_DEFINITION_SHAPE('', '#149);
#498 = PRODUCT_DEFINITION_SHAPE('', '#150);
#499 = PRODUCT_DEFINITION_SHAPE('', '#151);
#500 = PRODUCT_DEFINITION_SHAPE('', '#152);
#501 = PRODUCT_DEFINITION_SHAPE('', '#153);
#502 = PRODUCT_DEFINITION_SHAPE('', '#154);
#503 = PRODUCT_DEFINITION_SHAPE('', '#155);
#504 = PRODUCT_DEFINITION_SHAPE('', '#156);
#505 = PRODUCT_DEFINITION_SHAPE('', '#157);
#506 = PRODUCT_DEFINITION_SHAPE('', '#158);
#507 = PRODUCT_DEFINITION_SHAPE('', 'finished stratum extent', #521);
#508 = COMPONENT_DEFINITION('', 'interconnect module', #262, #575);
#509 = COMPONENT_DEFINITION('', 'interconnect module edge', #262, #576);
#510 = COMPONENT_DEFINITION('', 'non-plated thru-hole', #262, #576);
#511 = COMPONENT_DEFINITION('', 'non-plated thru-hole', #262, #576);
#512 = COMPONENT_DEFINITION('', 'non-plated thru-hole', #262, #576);
#513 = COMPONENT_DEFINITION('', 'non-plated thru-hole', #262, #576);
#514 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#515 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#516 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#517 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#518 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#519 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#520 = COMPONENT_DEFINITION('', 'plated thru-hole', #262, #576);
#521 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #407, #408);
#522 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #407, #407);
#523 = PRODUCT_DEFINITION_RELATIONSHIP('', 'inter stratum extent',
    '', #408, #408);

```

```

#524 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #546, #508);
#525 = PRODUCT_DEFINITION_RELATIONSHIP('', 'instantiated part',
    '', #431, #432);
#526 = PRODUCT_DEFINITION_RELATIONSHIP('', 'used package',
    '', #433, #431);
#527 = PRODUCT_DEFINITION_RELATIONSHIP('', 'design usage',
    '', #545, #547);
#528 = PRODUCT_DEFINITION_RELATIONSHIP('', 'design usage',
    '', #546, #548);
#529 = DOCUMENT_RELATIONSHIP('', '', #535, #531);
#530 = DOCUMENT_RELATIONSHIP('', '', #536, #532);
#531 = DOCUMENT('Preliminary', '', '', #533);
#532 = DOCUMENT('2.0', '', '', #534);
#533 = DOCUMENT_TYPE('design specification');
#534 = DOCUMENT_TYPE('language reference manual');
#535 = EE_SPECIFICATION('URLhttp://www.pdes.org', 'EMpilot Working Paper',
    '', #533);
#536 = EE_SPECIFICATION('URLhttp://www.intermedius.com',
    'Intermediate Data Format Specification', '', #534);
#537 = COORDINATED_UNIVERSAL_TIME_OFFSET(0, 0, .BEHIND.);
#538 = COORDINATED_UNIVERSAL_TIME_OFFSET(0, 0, .BEHIND.);
#539 = LOCAL_TIME(0, 0, 0., #537);
#540 = LOCAL_TIME(8, 48, 18., #538);
#541 = CALENDAR_DATE(1993, 3, 1);
#542 = CALENDAR_DATE(2000, 10, 2);
#543 = DATE_AND_TIME(#541, #539);
#544 = DATE_AND_TIME(#542, #540);
#545 = PHYSICAL_UNIT('', 'assembly module', #263, #578);
#546 = PHYSICAL_UNIT('', 'interconnect module', #262, #578);
#547 = PHYSICAL_UNIT('', 'assembly module', #263, #579);
#548 = PHYSICAL_UNIT('', 'interconnect module', #262, #579);
#549 = DESCRIPTIVE_REPRESENTATION_ITEM('plated passage', 'true');
#550 = DESCRIPTIVE_REPRESENTATION_ITEM('plated passage', 'false');
#551 = DESCRIPTIVE_REPRESENTATION_ITEM('feature of size', 'false');
#552 = DESCRIPTIVE_REPRESENTATION_ITEM('',
    'Design organization archival data set');
#553 = DESCRIPTIVE_REPRESENTATION_ITEM('',
    'Both the interconnect_module and the interconnect_module_component
    share the same geometric_curve_set');
#554 = DESCRIPTIVE_REPRESENTATION_ITEM('',
    'Added a representation_context
    and a representation that were not in the mapping tables, so the
    text_usage on an ee_document will pass WR1 for
    descriptive_representation_item');
#555 = DESCRIPTIVE_REPRESENTATION_ITEM('',

```



```

    'Added a representation_context
    with a context_type of "registered description" that was not in
    the mapping tables, so the context_of_items on a
    registered_case_style will pass the required attributes check');
#556 = DESCRIPTIVE_REPRESENTATION_ITEM('case style code',
    '8FP_28_MANUAL_0.058');
#557 = DESCRIPTIVE_REPRESENTATION_ITEM('manufacturing','');
#558 = DESCRIPTIVE_REPRESENTATION_ITEM('nominal material condition','');
#559 = DESCRIPTIVE_REPRESENTATION_ITEM('design','');
#560 = DESCRIPTIVE_REPRESENTATION_ITEM('design profile','');
#561 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#562 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#563 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','false');
#564 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#565 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#566 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#567 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#568 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#569 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#570 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#571 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#572 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#573 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#574 = DESCRIPTIVE_REPRESENTATION_ITEM('placement fixed','true');
#575 = PRODUCT_DEFINITION_CONTEXT('physical occurrence',#586,
    'as designed');
#576 = PRODUCT_DEFINITION_CONTEXT('layout occurrence',#586,
    'as designed');
#577 = PRODUCT_DEFINITION_CONTEXT('template definition',#586,
    'as designed');
#578 = PRODUCT_DEFINITION_CONTEXT('physical design usage',#586,
    'as designed');
#579 = PRODUCT_DEFINITION_CONTEXT('physical design',#586,
    'as designed');
#580 = PRODUCT_DEFINITION_CONTEXT('technology definition',#586,
    'as designed');
#581 = REPRESENTATION_CONTEXT('','');
#582 = REPRESENTATION_CONTEXT('','registered description');
#583 = REPRESENTATION_CONTEXT('','shape conditions');
#584 = PRODUCT_CONTEXT('',#586,'mechanical');
#585 = APPLICATION_PROTOCOL_DEFINITION('',
    'electronic_assembly_interconnect_and_packaging_design',
    2000,#586);
#586 = APPLICATION_CONTEXT('EM pilot');
ENDSEC;
END-ISO-10303-21;

```

- [1] *Cal Poly Task Team Extension Final Report*; Prepared by: Mary Kathryn Kennedy, Scott Madsen, Curtis H. Parks; National Institute of Standards and Technology, March 1988.
- [2] CHURCH, MIKE; HARVEY, EUGENE, *Conceptual Model of a Pcb*; produced by: EDIF PCB Technical Sub-Committee; Version 12; Electronic Industries Association, 1993.
- [3] ANSI/IPC D-275:1991 *Design Standard for Rigid Printed Boards and Rigid Printed Board Assemblies — The Institute for Interconnecting and Packaging Electronic Circuits*
- [4] ASME -Y14.5M:1994 *Dimensioning and Tolerancing — The American Society of Mechanical Engineers*; New York, NY
- [5] ANSI/IPC D-352:1985 *Electronic Design Data Description for Printed Board in Digital Form — The Institute for Interconnecting and Packaging Electronic Circuits*
- [6] NIST -183:1993 *Integration Definition for Function Modeling (IDEF0) — Federal Information Processing Standards*
- [7] GADIENT, A. J.; HINES, L. E.; et al., *Agility through Information Sharing: Results Achieved in a Production Setting*; Concurrent Engineering: Research and Applications Journal, pp. 101 - 111, volume 5, number 2; Technomic Publishing Co., Inc., June 1997.
- [8] IEC 60617-1:1985 *Graphical symbols for diagrams. — General information, general index. Cross-reference tables*
- [9] IEC 60617-2:1996 *Graphical symbols for diagrams. — Symbol elements, qualifying symbols and other symbols having general application*
- [10] IEC 60617-3:1996 *Graphical symbols for diagrams. — Conductors and connecting devices*
- [11] IEC 60617-4:1996 *Graphical symbols for diagrams. — Passive components*
- [12] IEC 60617-5:1996 *Graphical symbols for diagrams. — Semiconductors and electron tubes*
- [13] IEC 60617-6:1996 *Graphical symbols for diagrams. — Production and conversion of electrical energy*
- [14] IEC 60617-7:1996 *Graphical symbols for diagrams. — Switchgear, controlgear and protective devices*
- [15] IEC 60617-8:1996 *Graphical symbols for diagrams. — Measuring instruments, lamps and signalling devices*
- [16] IEC 60617-9:1996 *Graphical symbols for diagrams. — Telecommunications - Switching and peripheral equipment*
- [17] IEC 60617-10:1996 *Graphical symbols for diagrams. — Telecommunications - Transmission*

- [18] IEC 60617-11:1996 *Graphical symbols for diagrams. — Architectural and topographical installation plans and diagrams*
- [19] IEC 60617-12:1991 *Graphical symbols for diagrams. — Binary logical elements*
- [20] IEC 60617-13:1993 *Graphical symbols for diagrams — Analogue elements*
- [21] IEC 60848:1988 *Preparation of function charts for control systems*
- [22] IEC 61082-1:1991 *Preparation of documents used in electrotechnology — General requirements*
- [23] IEC 61082-2:1993 *Preparation of documents used in electrotechnology — Function-oriented diagrams*
- [24] IEC 61082-3:1993 *Preparation of documents used in electrotechnology — Connection diagrams, tables and lists*
- [25] IEC 61175:1993 *Designations for signals and connections*
- [26] IEC 61346-1:1996 *Industrial systems, installations and equipment and industrial products - Structuring principles and reference designations — Basic rules*
- [27] IEC ACDV 61690-2:1997 *EDIF 4 0 0*
- [28] ISO 286-1:1988 *ISO system of limits and fits — Bases of tolerances, deviations and fits.*
- [29] ISO 286-2:1988 *ISO system of limits and fits — Tables of standard tolerance grades and limit deviations for holes and shafts.*
- [30] ISO 406:1987 *Technical drawings — Tolerancing of linear and angular dimensions.*
- [31] ISO 1101:1995 *Geometrical Product Specifications (GPS) — Geometrical tolerancing — Generalities, definitions, symbols, indication on drawings*
- [32] ISO 5459:1981 *Technical drawings — Geometrical tolerancing — Datums and datum-systems for geometrical tolerances*
- [33] ISO 6523, Amendment 1:1994 *Data interchange — Structures for the identification of organizations*
- [34] ASME Y14.5.1-M:1994 *Mathematical Definition of Dimensioning and Tolerancing Principles — The American Society of Mechanical Engineers, New York, NY*
- [35] PEAK, R. S.; FULTON, R. E.; SITARAMAN, S. K., *Thermomechanical CAD/CAE Integration in the TIGER PWA Toolset*; edited by Suhir, E., et al.; EEP-Vol. 19-1; InterPACK'97, Kohala Coast, Hawaii, 957-962, *Advances in Electronic Packaging*, 1997.

- [36] ANSI/IPC D-350–D:1992 *Printed Board Description in Digital Form — The Institute for Interconnecting and Packaging Electronic Circuits*
- [37] ANSI/IPC D-351:1985 *Printed Board Drawings in Digital Form — The Institute for Interconnecting and Packaging Electronic Circuits*
- [38] TAMBURINI, D. R.; PEAK, R. S.; FULTON, R. E., *Driving PWA Thermomechanical Analysis from STEP AP210 Product Models*; 1997 ASME International Mechanical Engineering Congress and Exposition, Dallas, 1997.
- [39] ANSI/IPC T-50–E:1992 *Terms and Definitions for Interconnecting and Packaging Electronic Circuits — The Institute for Interconnecting and Packaging Electronic Circuits*
- [40] THURMAN, T. R., *Common Abstractions in Electronic Design and Relationship to ISO 10303 DIS 210 and other design data standards*; ISO TC184/SC4/JWG9/N26-96; June 1996.
- [41] TOLMIE, FREDERICK, *The Design Data Capture of Printed Circuit Boards and Assemblies; A Particular Approach Utilizing Geometric Dimensioning and Tolerancing*; Draft 10; Rockwell Avionics - Collins Divisions, April 1997.

Index

AIM EXPRESS expanded listing rules	4097–4098, 4110, 4119–4121, 4150, 4154–4155, 4159, 4165, 4168, 4196, 4227, 4248
absorbed_dose_data_element	
application object	96
ARM EXPRESS listing entities	4434
mapping table	1635
absorbed_dose_measure_with_unit	
AIM EXPRESS short listing entities	2962
mapping table	1635
action	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3830
AIM EXPRESS short listing entities	3474
mapping table	2798, 2800–2801
action_assigned_item	
AIM diagrams	4871
AIM EXPRESS expanded listing types	3810
AIM EXPRESS short listing types	2949
action_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3830
AIM EXPRESS short listing entities	3475
action_directive	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3830
AIM EXPRESS short listing entities	3475
mapping table	1681
action_method	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3830
mapping table	1679–1680
action_method_relationship	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3830
action_relationship	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3830
action_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3831
AIM EXPRESS short listing entities	3475
action_request_solution	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3831
action_request_status	

AIM diagrams	4869
AIM EXPRESS expanded listing entities	3831
AIM EXPRESS short listing entities	3476
mapping table	1685
action_status	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3831
activity_data_element	
application object	97
ARM EXPRESS listing entities	4434
mapping table	1635
activity_measure_with_unit	
AIM EXPRESS short listing entities	2963
mapping table	1635
acu_requires_security_classification	
AIM EXPRESS expanded listing rules	4071
acyclic_curve_replica	
AIM EXPRESS expanded listing functions	4252
acyclic_mapped_representation	
AIM EXPRESS expanded listing functions	4253
acyclic_point_replica	
AIM EXPRESS expanded listing functions	4253
acyclic_product_category_relationship	
AIM EXPRESS expanded listing functions	4254
acyclic_product_definition_relationship	
AIM EXPRESS expanded listing functions	4254
acyclic_solid_replica	
AIM EXPRESS expanded listing functions	4254
acyclic_surface_replica	
AIM EXPRESS expanded listing functions	4255
add_data_element	
ARM EXPRESS listing functions	4609
add_design_object_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3831
AIM EXPRESS short listing entities	2964
mapping table	1691
add_design_object_management_relationship	
application assertion	1150
application object	97
ARM diagrams	4666
ARM EXPRESS listing entities	4434
mapping table	1691
add_design_object_management_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4073
AIM EXPRESS short listing rules	3519

add_design_object_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3831
AIM EXPRESS short listing entities	2964
mapping table	1691
address	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3832
AIM EXPRESS short listing entities	3476
adjacent_stratum_surface_definition	
application assertion	1150
application object	100
ARM diagrams	4739
ARM EXPRESS listing entities	4435
mapping table	2046, 2910
adjacent_stratum_surface_definition_constraint	
AIM EXPRESS expanded listing rules	4074
AIM EXPRESS short listing rules	3522
adjacent_stratum_surface_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4075
AIM EXPRESS short listing rules	3523
adjacent_stratum_surface_embedded_component_surface_definition	
application assertion	1151, 1172
application object	98
ARM EXPRESS listing entities	4435
mapping table	2046, 2910
advanced_b_rep	
application object	101
ARM diagrams	4823
ARM EXPRESS listing entities	4435
mapping table	1365
advanced_brep_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3833
mapping table	1365
advanced_face	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3835
aggregate_connectivity_requirement	
application assertion	1151, 1191
application object	102
ARM diagrams	4684
ARM EXPRESS listing entities	4435
mapping table	1772, 1786
aggregate_connectivity_requirement_unique_constraint	
AIM EXPRESS expanded listing rules	4075

AIM EXPRESS short listing rules	3525
ahead_or_behind	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3810
ARM EXPRESS listing types	4400
altered_package	
application assertion	1152, 1277
application object	103
ARM diagrams	4819
ARM EXPRESS listing entities	4435
mapping table	1366, 1372, 2450
altered_package_terminal	
AIM EXPRESS short listing entities	2965
application assertion	1152
application object	105
ARM diagrams	4615
ARM EXPRESS listing entities	4436
mapping table	1370, 1376, 2454
altered_packaged_part	
application assertion	1152
application object	106
ARM diagrams	4615
ARM EXPRESS listing entities	4436
mapping table	1377, 2496
alternate_product	
application assertion	1152
application object	107
ARM diagrams	4620
ARM EXPRESS listing entities	4436
mapping table	1400, 1952
alternate_product_relationship	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	3835
AIM EXPRESS short listing entities	3476
mapping table	1400–1401, 1410, 1952–1953, 1963
alternate_select_product	
application assertion	1152
application object	108
ARM diagrams	4620
ARM EXPRESS listing entities	4436
mapping table	1400, 1952
amount_of_substance_data_element	
application object	109
ARM EXPRESS listing entities	4436
mapping table	1635
amount_of_substance_measure	

AIM diagrams	4896
AIM EXPRESS expanded listing types	3810
amount_of_substance_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3836
mapping table	1635
amount_of_substance_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3836
analytical_model	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3836
AIM EXPRESS short listing entities	2966
application assertion	1153–1154, 1288, 1308
application object	110
ARM diagrams	4616
ARM EXPRESS listing entities	4436
mapping table	1386, 1664
analytical_model_definition	
AIM EXPRESS short listing entities	2967
application assertion	1153
application object	113
ARM EXPRESS listing entities	4437
mapping table	1388, 2763
analytical_model_make_from_relationship	
AIM EXPRESS short listing entities	2969
application assertion	1153–1154
application object	114
ARM EXPRESS listing entities	4437
mapping table	1388
analytical_model_port	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3838
AIM EXPRESS short listing entities	2970
application assertion	1153–1154, 1191, 1196, 1284–1285, 1342
application object	115
ARM diagrams	4617
ARM EXPRESS listing entities	4437
mapping table	1389, 1392–1393, 1666, 1669, 1673
analytical_model_port_assignment	
application assertion	1153–1154
application object	117
ARM diagrams	4689
ARM EXPRESS listing entities	4437
mapping table	1825, 1832
analytical_model_port_assignment_select	

ARM diagrams	4857
analytical_model_port_make_from_relationship	
application assertion	1154
application object	118
ARM EXPRESS listing entities	4438
mapping table	1391
analytical_model_port_unique_constraint	
AIM EXPRESS expanded listing rules	4076
AIM EXPRESS short listing rules	3526
analytical_model_scalar_port	
application assertion	1341
application object	119
ARM diagrams	4617
ARM EXPRESS listing entities	4438
mapping table	1392, 1669
analytical_model_vector_port	
application object	120
ARM diagrams	4617
ARM EXPRESS listing entities	4438
mapping table	1392, 1669
analytical_model_vector_port_assignment	
ARM diagrams	4689
analytical_model_vector_port_assignment_constraint	
AIM EXPRESS short listing rules	3528
analytical_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3838
AIM EXPRESS short listing entities	2973
application assertion	1154, 1156, 1226, 1237, 1281, 1287, 1297, 1308
application object	121
ARM diagrams	4619
ARM EXPRESS listing entities	4438
mapping table	1399, 1825, 2586, 2687
angle_data_element	
application assertion	1166, 1169, 1218, 1234, 1291, 1321
application object	122
ARM diagrams	4659
ARM EXPRESS listing entities	4438
mapping table	1635
angle_relator	
AIM diagrams	4898
AIM EXPRESS expanded listing types	3810
ARM EXPRESS listing types	4400
angular_dimension_with_direction_vector	
application assertion	1154–1155
application object	122

ARM diagrams	4714
ARM EXPRESS listing entities	4438
mapping table	1833
angular_dimension_with_direction_vector_unique_constraint	
AIM EXPRESS expanded listing rules	4077
AIM EXPRESS short listing rules	3528
angular_dimension_with_orientation	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3838
AIM EXPRESS short listing entities	2974
mapping table	1833
angular_location	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3838
angular_size	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3838
AIM EXPRESS short listing entities	3477
mapping table	1834–1835
angular_size_dimension	
application object	124
ARM diagrams	4702
ARM EXPRESS listing entities	4438
mapping table	1834
angular_size_dimension_constraint	
AIM EXPRESS short listing rules	3530
angularity_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3839
AIM EXPRESS short listing entities	2975
application object	125
ARM diagrams	4691
ARM EXPRESS listing entities	4439
mapping table	1835
annotation_curve_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3839
mapping table	2925
annotation_fill_area	
AIM diagrams	4929
AIM EXPRESS expanded listing entities	3839
mapping table	2842
annotation_fill_area_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3839
AIM EXPRESS short listing entities	3477

mapping table	2842
annotation_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3839
annotation_symbol	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3840
annotation_symbol_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3840
annotation_text	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3840
annotation_text_character	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3841
annotation_text_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3841
apex	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3841
application object	126
ARM diagrams	4692
ARM EXPRESS listing entities	4439
mapping table	1835
application_context	
AIM diagrams	4859
AIM EXPRESS expanded listing entities	3841
AIM EXPRESS short listing entities	3477
application_context_element	
AIM diagrams	4859
AIM EXPRESS expanded listing entities	3841
application_context_requires_ap_definition	
AIM EXPRESS expanded listing rules	4078
AIM EXPRESS short listing rules	3530
application_protocol_definition	
AIM diagrams	4859
AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	3478
applied_action_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	2976
applied_approval_assignment	
AIM diagrams	4880

AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	2976
applied_certification_assignment	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	2977
applied_classification_assignment	
AIM diagrams	4885
AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	2977
applied_contract_assignment	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3842
AIM EXPRESS short listing entities	2977
applied_date_and_time_assignment	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2978
mapping table	2796
applied_date_assignment	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2979
applied_date_correlation	
AIM EXPRESS expanded listing functions	4256
applied_date_time_correlation	
AIM EXPRESS expanded listing functions	4257
applied_document_reference	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2979
mapping table	2250
applied_group_assignment	
AIM diagrams	4885
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2979
mapping table	1827
applied_organization_assignment	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2980
applied_person_and_organization_assignment	
AIM diagrams	4879
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2980
applied_person_assignment	

AIM diagrams	4879
AIM EXPRESS expanded listing entities	3843
AIM EXPRESS short listing entities	2981
applied_promissory_usage_in_product_model_assignment	
AIM EXPRESS short listing entities	2981
mapping table	1406, 1960, 2179, 2378, 2829
applied_security_classification_assignment	
AIM diagrams	4874
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	2982
approval	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	3478
mapping table	1613
approval_assigned_item	
AIM diagrams	4881
AIM EXPRESS expanded listing types	3812
AIM EXPRESS short listing types	2949
approval_assignment	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	3478
approval_date_time	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	3479
approval_person_organization	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	3479
approval_relationship	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
approval_requires_approval_date_time	
AIM EXPRESS expanded listing rules	4078
AIM EXPRESS short listing rules	3531
approval_requires_approval_person_organization	
AIM EXPRESS expanded listing rules	4078
AIM EXPRESS short listing rules	3532
approval_role	
AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
approval_role_constraint	
AIM EXPRESS short listing rules	3532
approval_status	

AIM diagrams	4880
AIM EXPRESS expanded listing entities	3844
AIM EXPRESS short listing entities	3479
mapping table	1613
approvals_are_assigned	
AIM EXPRESS expanded listing rules	4078
AIM EXPRESS short listing rules	3533
area_data_element	
application object	127
ARM EXPRESS listing entities	4439
mapping table	1635
area_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3812
area_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3845
mapping table	1635
area_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3845
array_placement_group	
application object	128
ARM diagrams	4627
ARM EXPRESS listing entities	4439
mapping table	1429, 1964, 2049
array_placement_group_component_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3845
AIM EXPRESS short listing entities	2982
mapping table	1429, 1964
array_placement_group_component_shape_aspect	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3845
AIM EXPRESS short listing entities	2983
mapping table	1964, 2049, 2135
assembled_feature_select	
ARM diagrams	4839
ARM EXPRESS listing types	4401
assembled_with_bonding	
application assertion	1155, 1167
application object	128
ARM diagrams	4636
ARM EXPRESS listing entities	4439
mapping table	1492
assembled_with_fasteners	

application assertion	1155
application object	131
ARM diagrams	4636
ARM EXPRESS listing entities	4439
mapping table	1492
assembly_alternate_product	
application assertion	1155
application object	132
ARM diagrams	4620
ARM EXPRESS listing entities	4439
mapping table	1401, 1953
assembly_alternate_product_constraint	
AIM EXPRESS short listing rules	3533
assembly_bond_definition	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3846
AIM EXPRESS short listing entities	2983
application assertion	1155–1156, 1200
application object	133
ARM diagrams	4654
ARM EXPRESS listing entities	4440
mapping table	1603
assembly_component	
application assertion ... 1156–1157, 1162–1163, 1175, 1177–1179, 1204, 1206, 1254, 1262, 1296, 1317, 1324, 1328, 1340	
application object	135
ARM diagrams	4624
ARM EXPRESS listing entities	4440
mapping table	1411, 1493, 2050
assembly_component_2d_shape	
application assertion	1156, 1173
application object	138
ARM diagrams	4650
ARM EXPRESS listing entities	4440
mapping table	1573, 2318
assembly_component_3d_shape	
application assertion	1156, 1174
application object	139
ARM diagrams	4650
ARM EXPRESS listing entities	4440
mapping table	1574, 2319
assembly_component_non_planar_shape	
application assertion	1156, 1174
application object	140
ARM EXPRESS listing entities	4441
mapping table	2321

assembly_component_placement_link	
application assertion	1157, 1265
application object	140
ARM EXPRESS listing entities	4441
mapping table	1429, 1965
assembly_component_usage	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	3846
AIM EXPRESS short listing entities	3480
mapping table	1401, 1444, 1494, 1953, 1957, 1986, 2054, 2120, 2205–2206, 2604, 2610, 2913–2914
assembly_component_usage_substitute	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	3846
AIM EXPRESS short listing entities	3480
assembly_composition_relationship	
application assertion	1157, 1301, 1339
application object	141
ARM diagrams	4621
ARM EXPRESS listing entities	4441
mapping table	1401, 1494, 1953, 2054
assembly_composition_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4079
AIM EXPRESS short listing rules	3534
assembly_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3847
AIM EXPRESS short listing entities	2985
mapping table	1536, 1550
assembly_group_component	
application assertion	1155, 1157, 1236, 1256, 1339
application object	143
ARM diagrams	4627
ARM EXPRESS listing entities	4441
mapping table	1430, 1966, 2056
assembly_group_component_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3847
AIM EXPRESS short listing entities	2986
mapping table	1430, 1966, 2056
assembly_group_component_definition_placement_link	
AIM EXPRESS short listing entities	2988
mapping table	1429, 1965
assembly_group_component_shape_aspect	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3847

AIM EXPRESS short listing entities	2990
mapping table	1966, 2056
assembly_group_component_shape_aspect_placement_link	
AIM EXPRESS short listing entities	2990
mapping table	1965
assembly_joint	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3848
AIM EXPRESS short listing entities	2992
application assertion	1157–1159, 1162–1163, 1172–1173, 1204, 1296
application object	144
ARM diagrams	4636
ARM EXPRESS listing entities	4442
mapping table	1492, 1496, 2371
assembly_material_composition_relationship	
application assertion	1160
application object	147
ARM diagrams	4622
ARM EXPRESS listing entities	4442
mapping table	1403, 1501, 1955, 2822
assembly_module	
application assertion	1157, 1161, 1243, 1272
application object	149
ARM diagrams	4805
ARM EXPRESS listing entities	4442
mapping table	1503, 1998
assembly_module_component	
application assertion	1160
application object	150
ARM diagrams	4623
ARM EXPRESS listing entities	4442
mapping table	1413
assembly_module_component_terminal	
application assertion	1159–1160
application object	151
ARM diagrams	4635
ARM EXPRESS listing entities	4442
mapping table	1503
assembly_module_interface_terminal	
AIM EXPRESS short listing entities	2995
application object	152
ARM EXPRESS listing entities	4442
mapping table	1540
assembly_module_join_terminal	
AIM EXPRESS short listing entities	2995
application assertion	1161

application object	152
ARM EXPRESS listing entities	4443
mapping table	1540
assembly_module_macro_component	
application assertion	1161
application object	153
ARM EXPRESS listing entities	4443
mapping table	1414, 1505
assembly_module_macro_component_join_terminal	
AIM EXPRESS short listing entities	2996
application assertion	1161
application object	154
ARM EXPRESS listing entities	4443
mapping table	1415, 1506
assembly_module_or_assembly_group_component	
ARM EXPRESS listing types	4401
assembly_module_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3849
AIM EXPRESS short listing entities	2996
application assertion	1160–1161, 1352
application object	155
ARM diagrams	4645
ARM EXPRESS listing entities	4443
mapping table	1540, 1549, 1887, 2548, 2571
assembly_module_usage_view	
application assertion	1160–1162
application object	157
ARM diagrams	4646
ARM EXPRESS listing entities	4443
mapping table	1542
assembly_module_usage_view_connector_relationship	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3850
AIM EXPRESS short listing entities	2998
application assertion	1161–1162
application object	160
ARM diagrams	4776
ARM EXPRESS listing entities	4444
mapping table	1545
assembly_module_usage_view_connector_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4081
AIM EXPRESS short listing rules	3536
assembly_or_interconnect_design_object_select	
ARM diagrams	4628
ARM EXPRESS listing types	4401

assembly_requirement_allocation	
application assertion	1162–1163, 1300
application object	161
ARM diagrams	4646
ARM EXPRESS listing entities	4444
mapping table	1560, 2236
assembly_requirement_allocation_constraint	
AIM EXPRESS short listing rules	3539
assembly_separation_requirement_allocation	
application assertion	1163
application object	162
ARM diagrams	4647
ARM EXPRESS listing entities	4444
mapping table	1561, 2238
assembly_separation_requirement_allocation_constraint	
AIM EXPRESS short listing rules	3540
assembly_shape_is_defined	
AIM EXPRESS expanded listing functions	4259
assembly_technology_specification	
application object	163
ARM diagrams	4670
ARM EXPRESS listing entities	4444
mapping table	1607
assigned_product_category_correlation	
ARM EXPRESS listing functions	4609
associated_surface	
AIM EXPRESS expanded listing functions	4259
auxiliary_characteristic_dimension_representation	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3850
AIM EXPRESS short listing entities	3000
mapping table	1867
axis1_placement	
AIM diagrams	4905
AIM EXPRESS expanded listing entities	3850
axis2_placement	
AIM diagrams	4905
AIM EXPRESS expanded listing types	3812
axis2_placement_2d	
AIM diagrams	4905
AIM EXPRESS expanded listing entities	3851
mapping table	1835, 2833
axis2_placement_3d	
AIM diagrams	4905
AIM EXPRESS expanded listing entities	3851
mapping table	1835, 2833

axis_placement	
application assertion ...	1154, 1163, 1196, 1211–1212, 1216, 1266–1267, 1270, 1322, 1349, 1353, 1356
application object	163
ARM diagrams	4693
ARM EXPRESS listing entities	4444
mapping table	1835, 2833
axis_placement_2d	
application assertion	1208, 1213, 1229, 1314, 1333, 1336, 1357–1358
application object	164
ARM diagrams	4693
ARM EXPRESS listing entities	4444
mapping table	1835, 2833
axis_placement_3d	
application assertion	1167, 1173, 1179, 1207, 1214, 1288, 1332, 1335, 1337, 1357–1359
application object	165
ARM diagrams	4693
ARM EXPRESS listing entities	4444
mapping table	1835, 2833
axis_placement_or_cto2d	
ARM EXPRESS listing types	4401
axis_placement_or_datum_based_vector_orientation_select	
ARM diagrams	4714
ARM EXPRESS listing types	4401
b_rep_2d	
ARM diagrams	4823
b_spline_curve	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	3852
b_spline_curve_form	
AIM diagrams	4915
AIM EXPRESS expanded listing types	3812
b_spline_curve_with_knots	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	3852
b_spline_surface	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	3853
b_spline_surface_form	
AIM diagrams	4910
AIM EXPRESS expanded listing types	3812
b_spline_surface_with_knots	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	3853
bag_to_set	
AIM EXPRESS expanded listing functions	4259

application interpreted constructs	4340
bare_die	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3854
AIM EXPRESS short listing entities	3000
application assertion	1163–1166, 1271, 1284
application object	165
ARM diagrams	4657
ARM EXPRESS listing entities	4445
mapping table	1618
bare_die_bottom_surface	
AIM EXPRESS short listing entities	3002
application assertion	1164
application object	168
ARM EXPRESS listing entities	4445
mapping table	1623
bare_die_component	
application assertion	1164, 1229
application object	169
ARM diagrams	4623
ARM EXPRESS listing entities	4445
mapping table	1417
bare_die_component_or_packaged_component	
ARM EXPRESS listing types	4401
bare_die_component_terminal	
application assertion	1158, 1164
application object	169
ARM diagrams	4637
ARM EXPRESS listing entities	4445
mapping table	1507
bare_die_edge_segment_surface	
AIM EXPRESS short listing entities	3002
application assertion	1164–1165
application object	170
ARM EXPRESS listing entities	4445
mapping table	1624, 1627
bare_die_edge_surface	
AIM EXPRESS short listing entities	3004
application assertion	1165
application object	171
ARM EXPRESS listing entities	4446
mapping table	1626
bare_die_surface	
AIM EXPRESS short listing entities	3004
application assertion	1166
application object	171

ARM EXPRESS listing entities	4446
mapping table	1627
bare_die_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3854
AIM EXPRESS short listing entities	3005
application assertion	1164–1165, 1229, 1352
application object	172
ARM diagrams	4658
ARM EXPRESS listing entities	4446
mapping table	1628, 1887, 2548, 2571
bare_die_terminal_or_packaged_part_terminal	
ARM EXPRESS listing types	4402
bare_die_terminal_surface_constituent_relationship	
application assertion	1165–1166
application object	173
ARM EXPRESS listing entities	4446
mapping table	1629
bare_die_top_surface	
AIM EXPRESS short listing entities	3006
application assertion	1166
application object	173
ARM EXPRESS listing entities	4446
mapping table	1629
bare_die_unique_constraint	
AIM EXPRESS expanded listing rules	4083
AIM EXPRESS short listing rules	3540
base_axis	
AIM EXPRESS expanded listing functions	4260
bevel_edge_feature_shape	
application assertion	1166
application object	174
ARM diagrams	4674
ARM EXPRESS listing entities	4446
mapping table	1729
bevel_edge_feature_shape_constraint	
AIM EXPRESS short listing rules	3542
bezier_curve	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	3854
bezier_surface	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	3854
blind_passage_template	
application assertion	1166, 1193
application object	176

ARM diagrams	4672
ARM EXPRESS listing entities	4447
mapping table	1730
blind_via	
application object	177
ARM diagrams	4739
ARM EXPRESS listing entities	4447
mapping table	2057
block	
AIM diagrams	4924
AIM EXPRESS expanded listing entities	3854
application object	178
ARM diagrams	4690
ARM EXPRESS listing entities	4447
mapping table	1657
bond_assembly_position	
application assertion	1167
application object	179
ARM diagrams	4650
ARM EXPRESS listing entities	4447
mapping table	1575, 2322
bond_category	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3854
AIM EXPRESS short listing entities	3007
mapping table	1720
bonded_conductive_base_blind_via	
application assertion	1166
application object	180
ARM diagrams	4739
ARM EXPRESS listing entities	4447
mapping table	2058
boolean_choose	
AIM EXPRESS expanded listing functions	4260
boolean_csg_operand	
ARM diagrams	4663
ARM EXPRESS listing types	4402
boolean_csg_result	
application assertion	1167–1168
application object	181
ARM diagrams	4663
ARM EXPRESS listing entities	4447
mapping table	1657, 2833
boolean_operand	
AIM diagrams	4913
AIM EXPRESS expanded listing types	3812

boolean_operator	
AIM diagrams	4913
AIM EXPRESS expanded listing types	3813
ARM diagrams	4663
ARM EXPRESS listing types	4402
mapping table	1657, 2833
boolean_property	
ARM diagrams	4660
ARM EXPRESS listing types	4402
boolean_result	
AIM diagrams	4913
AIM EXPRESS expanded listing entities	3855
mapping table	1657, 2833
bound_volume_shape	
application assertion	1167, 1273, 1348, 1357–1359
application object	182
ARM diagrams	4823
ARM EXPRESS listing entities	4447
mapping table	2835
boundary_based_position_tolerance	
application assertion	1168
application object	182
ARM diagrams	4694
ARM EXPRESS listing entities	4448
mapping table	1835
boundary_curve	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	3855
boundary_size_characteristic	
application assertion	1168
application object	184
ARM diagrams	4694
ARM EXPRESS listing entities	4448
mapping table	1836
boundary_size_characteristic_constraint	
AIM EXPRESS expanded listing rules	4083
AIM EXPRESS short listing rules	3543
boundary_zone_definition_with_specified_size	
application assertion	1168
application object	185
ARM diagrams	4700
ARM EXPRESS listing entities	4448
mapping table	1837
boundary_zone_definition_with_specified_size_constraint	
AIM EXPRESS short listing rules	3544
bounded_curve	

AIM diagrams	4911
AIM EXPRESS expanded listing entities	3855
bounded_pcurve	
AIM diagrams	4911
AIM EXPRESS expanded listing entities	3855
bounded_surface	
AIM diagrams	4909
AIM EXPRESS expanded listing entities	3855
bounded_surface_curve	
AIM diagrams	4912
AIM EXPRESS expanded listing entities	3855
bounding_curve	
application assertion	1168, 1187
application object	186
ARM diagrams	4824
ARM EXPRESS listing entities	4448
mapping table	2835
box_characteristic_select	
AIM diagrams	4937
AIM EXPRESS expanded listing types	3813
box_domain	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3856
box_height	
AIM diagrams	4937
AIM EXPRESS expanded listing types	3813
box_rotate_angle	
AIM diagrams	4936
AIM EXPRESS expanded listing types	3813
box_slant_angle	
AIM diagrams	4936
AIM EXPRESS expanded listing types	3813
box_width	
AIM diagrams	4927
AIM EXPRESS expanded listing types	3813
boxed_half_space	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3856
brep_with_voids	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	3856
build_2axes	
AIM EXPRESS expanded listing functions	4261
build_axes	
AIM EXPRESS expanded listing functions	4261
buried_via	

application object	189
ARM diagrams	4739
ARM EXPRESS listing entities	4448
mapping table	2058
bus_element_link	
AIM EXPRESS short listing entities	3008
application assertion	1169
application object	190
ARM EXPRESS listing entities	4448
mapping table	1758
bus_element_relationship	
ARM diagrams	4679
bus_element_relationship_constraint	
AIM EXPRESS expanded listing rules	4083
bus_element_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4084
bus_structural_definition	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	3856
AIM EXPRESS short listing entities	3009
application assertion	1169, 1238–1239
application object	190
ARM diagrams	4679
ARM EXPRESS listing entities	4449
mapping table	1759
bus_structural_definition_or_node	
ARM diagrams	4679
ARM EXPRESS listing types	4402
bus_structural_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4085
AIM EXPRESS short listing rules	3544
cable_component	
AIM EXPRESS short listing entities	3010
application object	192
ARM EXPRESS listing entities	4449
mapping table	1418
calendar_date	
AIM diagrams	4883
AIM EXPRESS expanded listing entities	3857
mapping table	1612
capacitance_data_element	
application object	193
ARM EXPRESS listing entities	4449
mapping table	1635
capacitance_measure_with_unit	
AIM EXPRESS short listing entities	3011

mapping table	1635
cartesian_coordinate_system	
application assertion	1169, 1323
application object	194
ARM diagrams	4825
ARM EXPRESS listing entities	4449
mapping table	2835
cartesian_coordinate_system_constraint	
AIM EXPRESS expanded listing rules	4086
AIM EXPRESS short listing rules	3545
cartesian_point	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	3857
application assertion	1163–1164, 1169, 1172, 1198, 1233–1234, 1258, 1277, 1288, 1290, 1297–1299, 1303, 1348, 1358
application object	195
ARM diagrams	4825
ARM EXPRESS listing entities	4449
mapping table	2838
cartesian_transformation_operator	
AIM diagrams	4903
AIM EXPRESS expanded listing entities	3857
mapping table	2838
cartesian_transformation_operator_2d	
AIM diagrams	4903
AIM EXPRESS expanded listing entities	3857
application assertion	1169, 1172
application object	195
ARM diagrams	4825
ARM EXPRESS listing entities	4449
mapping table	2838
cartesian_transformation_operator_2d_constraint	
AIM EXPRESS expanded listing rules	4086
AIM EXPRESS short listing rules	3547
cartesian_transformation_operator_3d	
AIM diagrams	4903
AIM EXPRESS expanded listing entities	3858
celsius_temperature_data_element	
application object	197
ARM EXPRESS listing entities	4450
mapping table	1635
celsius_temperature_measure_with_unit	
mapping table	1635
centre_axis	
application assertion	1218, 1241
application object	198

ARM diagrams	4692
ARM EXPRESS listing entities	4450
mapping table	1838
centre_of_symmetry	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3858
AIM EXPRESS short listing entities	3481
application object	198
ARM diagrams	4692
ARM EXPRESS listing entities	4450
mapping table	1838
centre_of_symmetry_constraint	
AIM EXPRESS short listing rules	3548
centre_plane	
application assertion	1241
application object	199
ARM diagrams	4692
ARM EXPRESS listing entities	4450
mapping table	1838
centre_point	
application assertion	1242
application object	200
ARM diagrams	4692
ARM EXPRESS listing entities	4450
mapping table	1838
centre_type	
ARM diagrams	4703
ARM EXPRESS listing types	4402
certification	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3858
application assertion	1170
application object	200
ARM diagrams	4835
ARM EXPRESS listing entities	4450
mapping table	2873
certification_assigned_item	
AIM diagrams	4873
AIM EXPRESS expanded listing types	3813
AIM EXPRESS short listing types	2950
certification_assignment	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3858
certification_requires_date_or_date_and_time	
AIM EXPRESS expanded listing rules	4086
AIM EXPRESS short listing rules	3548

certification_type	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3858
AIM EXPRESS short listing entities	3481
certification_unique_constraint	
AIM EXPRESS expanded listing rules	4087
AIM EXPRESS short listing rules	3549
change	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3858
AIM EXPRESS short listing entities	3012
change_design_object_management_relationship	
application assertion	1170
application object	201
ARM diagrams	4666
ARM EXPRESS listing entities	4450
mapping table	1691
change_design_object_management_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4090
AIM EXPRESS short listing rules	3550
change_from_design_object_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3859
AIM EXPRESS short listing entities	3013
mapping table	1691, 1741, 1746–1748, 1754–1755
change_from_design_object_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3859
AIM EXPRESS short listing entities	3014
mapping table	1691
change_order	
application assertion	1170–1171
application object	203
ARM diagrams	4664
ARM EXPRESS listing entities	4451
mapping table	1679
change_request	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3859
AIM EXPRESS short listing entities	3015
application assertion	1171
application object	203
ARM diagrams	4665
ARM EXPRESS listing entities	4451
mapping table	1679
change_request_item	

AIM diagrams	4870
AIM EXPRESS expanded listing types	3813
AIM EXPRESS short listing types	2950
change_request_unique_constraint	
AIM EXPRESS expanded listing rules	4091
AIM EXPRESS short listing rules	3554
change_to_design_object_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3859
AIM EXPRESS short listing entities	3015
mapping table	1741, 1746–1748, 1754–1755
change_to_design_object_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3860
AIM EXPRESS short listing entities	3016
character_glyph_font_usage	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	3860
character_glyph_style_outline	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	3860
character_glyph_style_stroke	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	3860
character_glyph_symbol	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	3860
character_glyph_symbol_outline	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	3861
character_spacing_select	
AIM diagrams	4927
AIM EXPRESS expanded listing types	3813
character_style_select	
AIM diagrams	4933
AIM EXPRESS expanded listing types	3814
characteristic	
application assertion	1171, 1193, 1225, 1266, 1271, 1281, 1283, 1287, 1297, 1322
application object	205
ARM diagrams	4660
ARM EXPRESS listing entities	4451
mapping table	1635
characteristic_constraint	
AIM EXPRESS short listing rules	3556
characteristic_data_set	
application assertion	1171

application object	206
ARM EXPRESS listing entities	4451
mapping table	1641, 2868
characteristic_data_table	
application assertion	1172, 1325
application object	206
ARM EXPRESS listing entities	4451
mapping table	1641, 2868
characteristic_type	
AIM EXPRESS short listing entities	3017
characteristic_unique_constraint	
AIM EXPRESS expanded listing rules	4092
characterized_definition	
AIM diagrams	4865
AIM EXPRESS expanded listing types	3814
characterized_material_property	
AIM diagrams	4868
AIM EXPRESS expanded listing types	3814
characterized_object	
AIM diagrams	4863
AIM EXPRESS expanded listing entities	3861
AIM EXPRESS short listing entities	3481
mapping table	1453, 1459, 2299, 2304, 2436, 2884
characterized_product_category	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	3861
AIM EXPRESS short listing entities	3018
mapping table	2444, 2666, 2761
characterized_product_definition	
AIM diagrams	4865
AIM EXPRESS expanded listing types	3814
circle	
AIM diagrams	4914
AIM EXPRESS expanded listing entities	3861
circular_area	
application assertion	1172
application object	207
ARM EXPRESS listing entities	4451
mapping table	2839
circular_runout_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3862
AIM EXPRESS short listing entities	3020
application object	207
ARM diagrams	4691
ARM EXPRESS listing entities	4451

mapping table	1838
circularity_tolerance	
application object	209
ARM diagrams	4696
ARM EXPRESS listing entities	4452
mapping table	1838
classification_assigned_item	
AIM diagrams	4885
AIM EXPRESS expanded listing types	3814
AIM EXPRESS short listing types	2950
closed_curve	
application assertion	1233
application object	210
ARM diagrams	4824
ARM EXPRESS listing entities	4452
mapping table	2839
closed_shell	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3862
colour	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3862
colour_rgb	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3862
mapping table	1645
colour_specification	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3862
common_datum	
application assertion	1197
application object	210
ARM diagrams	4673
ARM EXPRESS listing entities	4452
mapping table	1838
compatible_dimension	
AIM EXPRESS expanded listing rules	4092
component	9
component_2d_edge_location	
application assertion	1172
application object	211
ARM diagrams	4651
ARM EXPRESS listing entities	4452
mapping table	1577, 2325
component_2d_embedded_location	
application assertion	1172

application object	212
ARM EXPRESS listing entities	4452
mapping table	2326
component_2d_location	
application assertion	1172–1173
application object	213
ARM diagrams	4651
ARM EXPRESS listing entities	4453
mapping table	1578, 2328
component_2d_stacked_location	
application assertion	1173
application object	215
ARM EXPRESS listing entities	4453
mapping table	1579
component_2d_surface_location	
application assertion	1173
application object	216
ARM EXPRESS listing entities	4453
mapping table	1580
component_3d_location	
application assertion	1173–1174
application object	217
ARM diagrams	4651
ARM EXPRESS listing entities	4453
mapping table	1581, 2329
component_assembly_2d_position	
application assertion	1173
application object	218
ARM diagrams	4652
ARM EXPRESS listing entities	4453
mapping table	1582, 2330
component_assembly_3d_position	
application assertion	1173–1174
application object	219
ARM diagrams	4652
ARM EXPRESS listing entities	4454
mapping table	1585, 2334
component_assembly_non_planar_position	
application assertion	1174
application object	220
ARM EXPRESS listing entities	4454
mapping table	2337
component_based_annotation_symbol_placement_relationship	
application assertion	1175
application object	221
ARM diagrams	4740

ARM EXPRESS listing entities	4454
mapping table	2059
component_based_annotation_text_relationship	
application assertion	1175
application object	222
ARM diagrams	4740
ARM EXPRESS listing entities	4454
mapping table	2060
component_based_message_type	
ARM diagrams	4740
ARM EXPRESS listing types	4402
component_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3866
AIM EXPRESS short listing entities	3021
mapping table ... 1405, 1411, 1413–1414, 1417–1419, 1424–1425, 1438, 1493, 1505, 1528, 1537, 2050, 2117, 2375, 2824	
component_definitioncomponent_definition	
mapping table	1555
component_external_reference	
application assertion	1176–1177, 1207, 1254, 1318, 1328
application object	224
ARM diagrams	4741
ARM EXPRESS listing entities	4454
mapping table	2062, 2259
component_external_reference_constraint	
AIM EXPRESS expanded listing rules	4093
AIM EXPRESS short listing rules	3558
component_feature	
AIM EXPRESS short listing entities	3029
application assertion	1151, 1163, 1175–1178, 1207, 1254, 1262, 1318, 1324, 1328
application object	225
ARM diagrams	4638
ARM EXPRESS listing entities	4455
mapping table	1508, 1521, 1898, 2063, 2075
component_feature_external_reference	
application assertion	1176–1177, 1207, 1255, 1318, 1328
application object	227
ARM diagrams	4741
ARM EXPRESS listing entities	4455
mapping table	2069, 2260
component_feature_external_reference_constraint	
AIM EXPRESS short listing rules	3559
component_feature_external_reference_unique_constraint	
AIM EXPRESS expanded listing rules	4094
AIM EXPRESS short listing rules	3560

component_feature_make_from_relationship	
application assertion	1176
application object	228
ARM diagrams	4639
ARM EXPRESS listing entities	4455
mapping table	1513, 2069
component_feature_or_stratum_surface	
ARM EXPRESS listing types	4403
component_feature_relationship	
AIM EXPRESS short listing entities	3029
application assertion	1176, 1179
application object	229
ARM diagrams	4639
ARM EXPRESS listing entities	4455
mapping table	1513, 2070
component_feature_to_layout_feature_relationship	
application assertion	1176–1178
application object	230
ARM diagrams	4740
ARM EXPRESS listing entities	4455
mapping table	2070
component_feature_to_layout_feature_relationship_constraint	
AIM EXPRESS expanded listing rules	4094
AIM EXPRESS short listing rules	3562
component_feature_to_physical_usage_view_assignment	
application assertion	1178
application object	231
ARM diagrams	4631
ARM EXPRESS listing entities	4455
mapping table	1470, 1988
component_feature_unique_constraint	
AIM EXPRESS expanded listing rules	4095
AIM EXPRESS short listing rules	3563
component_functional_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3866
AIM EXPRESS short listing entities	3030
mapping table	1767
component_functional_unit	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	3867
AIM EXPRESS short listing entities	3031
mapping table	1759
component_group_assignment	
application assertion	1178
application object	232

ARM diagrams	4627
ARM EXPRESS listing entities	4456
mapping table	1446, 2260
component_interface_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3868
AIM EXPRESS short listing entities	3033
application assertion	1179
application object	233
ARM diagrams	4639
ARM EXPRESS listing entities	4456
mapping table ...	1479, 1508, 1514, 1521, 1524, 1535, 1755, 1781, 1793, 1808, 1898, 1960, 1977, 2001, 2006, 2031, 2063, 2074–2075, 2164, 2215, 2241, 2312, 2360, 2378, 2661, 2729, 2829, 2904
component_location	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3869
AIM EXPRESS short listing entities	3035
mapping table	1577–1581, 2325–2326, 2328–2329
component_make_from_relationship	
application assertion	1179
application object	234
ARM diagrams	4640
ARM EXPRESS listing entities	4456
mapping table	1515, 2074
component_mating_constraint_condition	
AIM EXPRESS short listing entities	3046
application object	235
ARM EXPRESS listing entities	4456
mapping table	1516, 2373
component_mounting_clearance_relationship	
application assertion	1179
application object	236
ARM diagrams	4640
ARM EXPRESS listing entities	4456
mapping table	1518
component_mounting_clearance_relationship_constraint	
AIM EXPRESS expanded listing rules	4096
AIM EXPRESS short listing rules	3565
component_mounting_feature	
AIM EXPRESS short listing entities	3047
application assertion	1159, 1179
application object	238
ARM EXPRESS listing entities	4456
mapping table	1519, 1898
component_non_planar_location	
application assertion	1174, 1179

application object	238
ARM EXPRESS listing entities	4457
mapping table	2339
component_or_component_feature	
ARM diagrams	4740
ARM EXPRESS listing types	4403
component_or_feature	
ARM diagrams	4647
ARM EXPRESS listing types	4403
component_overlap_relationship	
application assertion	1180
application object	239
ARM diagrams	4640
ARM EXPRESS listing entities	4457
mapping table	1519
component_overlap_relationship_constraint	
AIM EXPRESS expanded listing rules	4096
AIM EXPRESS short listing rules	3566
component_part_2d_geometric_representation_relationship	
application object	242
ARM diagrams	4653
ARM EXPRESS listing entities	4457
mapping table	1588, 2339
component_part_2d_geometric_representation_relationship_constraint	
AIM EXPRESS expanded listing rules	4096
AIM EXPRESS short listing rules	3567
component_part_2d_non_planar_geometric_representation_relationship	
application object	241
ARM EXPRESS listing entities	4457
mapping table	2342
component_part_3d_geometric_representation_relationship	
application object	243
ARM diagrams	4653
ARM EXPRESS listing entities	4457
mapping table	1591, 2344
component_part_3d_geometric_representation_relationship_constraint	
AIM EXPRESS short listing rules	3568
component_placement_restriction_assignment	
application assertion	1180–1181
application object	244
ARM diagrams	4628
ARM EXPRESS listing entities	4458
mapping table	1446, 1551, 2261
component_placement_restriction_assignment_constraint	
AIM EXPRESS expanded listing rules	4097
AIM EXPRESS short listing rules	3570

component_shape_aspect	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3873
AIM EXPRESS short listing entities	3049
mapping table	1899, 2050, 2082, 2085–2086, 2127, 2133, 2169, 2187, 2196, 2208
component_swap_specification	
application assertion	1181
application object	247
ARM diagrams	4675
ARM EXPRESS listing entities	4458
mapping table	1741, 1748
component_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3880
AIM EXPRESS short listing entities	3057
application assertion	1294
application object	248
ARM diagrams	4642
ARM EXPRESS listing entities	4458
mapping table	1479, 1503, 1507–1508, 1521, 1525, 1527, 1530, 1532–1533, 1755, 1778, 1781, 1793, 1807–1808, 1898, 1960, 1977, 1993, 2001, 2006, 2031, 2063, 2075, 2080, 2083, 2101, 2130–2131, 2139, 2151, 2164, 2172, 2209, 2215, 2241, 2312, 2360, 2378, 2661, 2729, 2825, 2829, 2904
component_terminal_to_assembly_module_interface_terminal_assignment	
application object	250
ARM EXPRESS listing entities	4458
mapping table	1471
component_terminal_to_assembly_module_interface_terminal_assignment_constraint	
AIM EXPRESS short listing rules	3573
component_terminal_to_assembly_module_interface_terminal_assignment_unique_constraint	
AIM EXPRESS short listing rules	3574
component_terminal_to_assembly_module_join_terminal_assignment	
application object	250
ARM EXPRESS listing entities	4458
mapping table	1473
component_terminal_to_assembly_module_join_terminal_assignment_constraint	
AIM EXPRESS short listing rules	3575
component_terminal_to_assembly_module_join_terminal_assignment_unique_constraint	
AIM EXPRESS short listing rules	3575
component_terminal_to_assembly_module_terminal_assignment	
ARM diagrams	4633
component_terminal_to_interconnect_module_interface_terminal_assignment	
application object	251
ARM EXPRESS listing entities	4459
mapping table	1989
component_terminal_to_interconnect_module_interface_terminal_assignment_constraint	

AIM EXPRESS short listing rules	3576
component_terminal_to_interconnect_module_interface_terminal_assignment_unique_constraint	
AIM EXPRESS short listing rules	3577
component_terminal_to_interconnect_module_join_terminal_assignment	
application object	251
ARM EXPRESS listing entities	4459
mapping table	1991
component_terminal_to_interconnect_module_join_terminal_assignment_constraint	
AIM EXPRESS short listing rules	3578
component_terminal_to_interconnect_module_join_terminal_assignment_unique_constraint	
AIM EXPRESS short listing rules	3579
component_terminal_to_interconnect_module_terminal_assignment	
ARM diagrams	4734
component_terminal_to_pca_terminal_assignment	
application object	252
ARM diagrams	4632
ARM EXPRESS listing entities	4459
mapping table	1474
component_terminal_to_pca_terminal_assignment_constraint	
AIM EXPRESS short listing rules	3580
component_terminal_to_pca_terminal_assignment_unique_constraint	
AIM EXPRESS expanded listing rules	4099
AIM EXPRESS short listing rules	3581
component_termination_passage	
application assertion	1181–1183, 1245, 1263
application object	253
ARM diagrams	4742
ARM EXPRESS listing entities	4459
mapping table	2078
component_termination_passage_and_contact_size_dependent_land	
application assertion	1181
application object	256
ARM diagrams	4771
ARM EXPRESS listing entities	4459
mapping table	2382
component_termination_passage_and_contact_size_dependent_non_functional_land	
application assertion	1182
application object	257
ARM diagrams	4771
ARM EXPRESS listing entities	4459
mapping table	2384
component_termination_passage_dependent_land	
application assertion	1182
application object	259
ARM diagrams	4743
ARM EXPRESS listing entities	4460

mapping table	2386
component_termination_passage_dependent_non_functional_land	
application assertion	1182–1183
application object	260
ARM diagrams	4772
ARM EXPRESS listing entities	4460
mapping table	2387
component_termination_passage_interface_terminal	
application assertion	1183
application object	261
ARM diagrams	4741
ARM EXPRESS listing entities	4460
mapping table	2078
component_termination_passage_join_terminal	
application assertion	1183, 1231, 1260
application object	262
ARM diagrams	4742
ARM EXPRESS listing entities	4460
mapping table	2080
component_termination_passage_template	
application assertion	1181, 1183–1184
application object	263
ARM diagrams	4731
ARM EXPRESS listing entities	4460
mapping table	1942, 2405
component_termination_passage_template_interface_terminal	
application assertion	1183
application object	264
ARM diagrams	4732
ARM EXPRESS listing entities	4460
mapping table	1942
component_termination_passage_template_join_terminal	
application assertion	1183–1184
application object	264
ARM diagrams	4732
ARM EXPRESS listing entities	4460
mapping table	1942
component_termination_passage_template_terminal	
application assertion	1175, 1184, 1350, 1352
application object	266
ARM diagrams	4732
ARM EXPRESS listing entities	4461
mapping table	1943
component_termination_passage_template_terminal_constraint	
AIM EXPRESS short listing rules	3582
component_termination_passage_template_terminal_unique_constraint	

AIM EXPRESS expanded listing rules	4100
AIM EXPRESS short listing rules	3583
composite_array_shape_aspect	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3880
AIM EXPRESS short listing entities	3069
mapping table	2548
composite_array_shape_aspect_link	
AIM EXPRESS short listing entities	3069
mapping table	2559
composite_curve	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	3881
application assertion	1184
application object	268
ARM diagrams	4824
ARM EXPRESS listing entities	4461
mapping table	2839
composite_curve_on_surface	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	3881
composite_curve_segment	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	3881
application assertion	1184
application object	269
ARM diagrams	4826
ARM EXPRESS listing entities	4461
mapping table	2839
composite_feature_locating_position_tolerance	
application assertion	1184
application object	270
ARM diagrams	4691
ARM EXPRESS listing entities	4461
mapping table	1838
composite_feature_relating_position_tolerance	
application assertion	1184
application object	270
ARM diagrams	4697
ARM EXPRESS listing entities	4461
mapping table	1839
composite_group_shape_aspect	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3881
AIM EXPRESS short listing entities	3070
mapping table	1840

composite_group_shape_element	
application object	272
ARM diagrams	4697
ARM EXPRESS listing entities	4461
mapping table	1840
composite_linear_profile_locating_tolerance	
application assertion	1185
application object	273
ARM diagrams	4691
ARM EXPRESS listing entities	4462
mapping table	1840
composite_linear_profile_refinement_tolerance	
application assertion	1185
application object	274
ARM diagrams	4693
ARM EXPRESS listing entities	4462
mapping table	1841
composite_shape_aspect	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3881
AIM EXPRESS short listing entities	3482
mapping table	1842, 1887, 2548, 2559
composite_shape_element	
application assertion	1323
application object	276
ARM diagrams	4697
ARM EXPRESS listing entities	4462
mapping table	1842
composite_shape_element_constraint	
AIM EXPRESS expanded listing rules	4100
AIM EXPRESS short listing rules	3585
composite_signal_property	
application assertion	1185
application object	277
ARM diagrams	4833
ARM EXPRESS listing entities	4462
mapping table	2868
composite_signal_property_constraint	
AIM EXPRESS short listing rules	3586
composite_signal_property_relationship	
AIM EXPRESS short listing entities	3071
application assertion	1185
application object	277
ARM diagrams	4834
ARM EXPRESS listing entities	4462
mapping table	2868

composite_surface_profile_locating_tolerance	
application assertion	1185
application object	278
ARM diagrams	4691
ARM EXPRESS listing entities	4462
mapping table	1842
composite_surface_profile_refinement_tolerance	
application assertion	1185–1186
application object	279
ARM diagrams	4696
ARM EXPRESS listing entities	4462
mapping table	1843
composite_unit_shape_aspect	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3882
AIM EXPRESS short listing entities	3072
mapping table	1844
composite_unit_shape_element	
application assertion	1216
application object	281
ARM diagrams	4697
ARM EXPRESS listing entities	4462
mapping table	1844
compound_characteristic	
application assertion	1186
application object	267
ARM EXPRESS listing entities	4461
mapping table	1642
compound_characteristic_definition	
application assertion	1171, 1186
ARM EXPRESS listing types	4403
compound_representation_item	
mapping table	1642
concentricity_and_coaxiality_tolerance	
application object	282
ARM diagrams	4691
ARM EXPRESS listing entities	4463
mapping table	1844
concentricity_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3882
AIM EXPRESS short listing entities	3073
mapping table	1844
conditional_reverse	
AIM EXPRESS expanded listing functions	4261
conductance_data_element	

application object	283
ARM EXPRESS listing entities	4463
mapping table	1643
conductance_measure_with_unit	
AIM EXPRESS short listing entities	3074
mapping table	1643
conductive_filled_area	
application assertion	1233, 1246, 1263
application object	284
ARM diagrams	4743
ARM EXPRESS listing entities	4463
mapping table	2082
conductive_interconnect_element	
application assertion	1186, 1212, 1293, 1295
application object	284
ARM diagrams	4745
ARM EXPRESS listing entities	4463
mapping table	2082
conductive_interconnect_element_terminal	
application assertion	1186, 1211, 1213, 1232, 1260
application object	286
ARM diagrams	4746
ARM EXPRESS listing entities	4463
mapping table	2083
conductive_interconnect_element_terminal_link	
AIM EXPRESS short listing entities	3074
application object	287
ARM EXPRESS listing entities	4464
mapping table	2014
conductive_interconnect_element_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4101
AIM EXPRESS short listing rules	3587
conductive_interconnect_element_with_pre_defined_transitions	
application assertion	1186–1187
application object	289
ARM diagrams	4745
ARM EXPRESS listing entities	4464
mapping table	2085
conductive_interconnect_element_with_pre_defined_transitions_constraint	
AIM EXPRESS short listing rules	3589
conductive_interconnect_element_with_user_defined_single_transition	
application assertion	1187–1188
application object	290
ARM diagrams	4746
ARM EXPRESS listing entities	4464
mapping table	2086

conductive_interconnect_element_with_user_defined_single_transition_constraint	
AIM EXPRESS short listing rules	3590
conductive_interconnect_terminal_or_interconnect_component_join_terminal	
ARM EXPRESS listing types	4403
conductor	
application assertion	1188, 1245, 1263
application object	292
ARM diagrams	4747
ARM EXPRESS listing entities	4464
mapping table	2091
configuration_controlled_message	
application assertion	1188, 1268
application object	294
ARM diagrams	4786
ARM EXPRESS listing entities	4464
mapping table	1486, 1702, 2008
configuration_controlled_message_constraint	
AIM EXPRESS short listing rules	3591
configuration_design	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3882
configuration_effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3882
AIM EXPRESS short listing entities	3482
mapping table	2725
configuration_item	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3882
AIM EXPRESS short listing entities	3483
mapping table	1718
configuration_item_requires_person_organization	
AIM EXPRESS expanded listing rules	4102
AIM EXPRESS short listing rules	3592
conic	
AIM diagrams	4914
AIM EXPRESS expanded listing entities	3883
conical_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	3883
conical_tolerance_zone_boundary	
application assertion	1189
application object	295
ARM diagrams	4699
ARM EXPRESS listing entities	4465
mapping table	1845

conical_tolerance_zone_boundary_and_surface_relationship	
application assertion	1188–1189
application object	296
ARM diagrams	4690
ARM EXPRESS listing entities	4465
mapping table	1845
conical_tolerance_zone_boundary_and_surface_relationship_constraint	
AIM EXPRESS short listing rules	3592
conical_tolerance_zone_boundary_definition	
application assertion	1189
application object	297
ARM diagrams	4700
ARM EXPRESS listing entities	4465
mapping table	1845
conical_tolerance_zone_boundary_definition_constraint	
AIM EXPRESS short listing rules	3593
conical_tolerance_zone_boundary_diametrical_size_characteristic	
application assertion	1189, 1242
application object	298
ARM diagrams	4698
ARM EXPRESS listing entities	4465
mapping table	1846
conical_tolerance_zone_boundary_diametrical_size_characteristic_constraint	
AIM EXPRESS short listing rules	3594
conical_tolerance_zone_definition	
application assertion	1189
application object	300
ARM diagrams	4701
ARM EXPRESS listing entities	4465
mapping table	1847
conical_tolerance_zone_definition_constraint	
AIM EXPRESS short listing rules	3595
connected	9
connected_area_component	
AIM EXPRESS short listing entities	3076
application assertion	1190
application object	301
ARM EXPRESS listing entities	4465
mapping table	2092
connected_edge_set	
AIM diagrams	4918
AIM EXPRESS expanded listing entities	3883
connected_face_set	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3883
connected_filled_area	

application assertion	1190
application object	301
ARM diagrams	4743
ARM EXPRESS listing entities	4465
mapping table	2093
connection	9
connection_zone	
application assertion ... 1161, 1165, 1184, 1186, 1190–1191, 1200, 1212, 1243, 1249, 1257, 1281, 1286–1287, 1311, 1354–1355, 1359	
application object	302
ARM diagrams	4826
ARM EXPRESS listing entities	4466
mapping table	2839
connection_zone_based_assembly_joint	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3883
AIM EXPRESS short listing entities	3077
application assertion	1190
application object	303
ARM diagrams	4695
ARM EXPRESS listing entities	4466
mapping table	1524
connection_zone_based_fabrication_joint	
AIM EXPRESS short listing entities	3078
application object	305
ARM EXPRESS listing entities	4466
mapping table	1723, 2015, 2094
connection_zone_constraint	
AIM EXPRESS expanded listing rules	4102
AIM EXPRESS short listing rules	3596
connection_zone_interface_plane_relationship	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	3884
AIM EXPRESS short listing entities	3079
application assertion	1165, 1190, 1280
application object	306
ARM diagrams	4829
ARM EXPRESS listing entities	4467
mapping table	1630, 2455, 2514
connection_zone_map_identification	
AIM EXPRESS short listing entities	3081
application assertion	1191, 1235
application object	307
ARM EXPRESS listing entities	4467
mapping table	2589
connection_zone_model_port_assignment	

application assertion	1191
application object	308
ARM diagrams	4663
ARM EXPRESS listing entities	4467
mapping table	1656, 2515, 2687
connectivity	9
connectivity_allocation	
application assertion	1191
application object	309
ARM diagrams	4684
ARM EXPRESS listing entities	4467
mapping table	1773, 1787
connectivity_allocation_constraint	
AIM EXPRESS expanded listing rules	4103
connectivity_allocation_to_physical_network	
application assertion	1191–1192
application object	310
ARM EXPRESS listing entities	4467
mapping table	1998
connectivity_allocation_unique_constraint	
AIM EXPRESS expanded listing rules	4104
connectivity_component	
ARM diagrams	4687
connectivity_sub_structure	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3884
ARM diagrams	4686
consistent_bus_structural_definition	
ARM EXPRESS listing functions	4610
consistent_part_template_shape_projection	
ARM diagrams	4800
consistent_physical_unit_shape_projection	
ARM diagrams	4805
consistent_stack_definition	
ARM EXPRESS listing functions	4610
consistent_stratum_shape_projection	
ARM diagrams	4841
constraining_element_definitions	
ARM diagrams	4839
ARM EXPRESS listing types	4404
constraint_occurrence	
application object	311
ARM diagrams	4821
ARM EXPRESS listing entities	4468
mapping table	2804
constraints_composite_curve_on_surface	

AIM EXPRESS expanded listing functions	4262
constraints_geometry_shell_based_surface_model	
AIM EXPRESS expanded listing functions	4262
constraints_geometry_shell_based_wireframe_model	
AIM EXPRESS expanded listing functions	4263
constraints_param_b_spline	
AIM EXPRESS expanded listing functions	4264
constraints_rectangular_composite_surface	
AIM EXPRESS expanded listing functions	4264
contact_size_dependent_land	
application assertion	1192
application object	311
ARM diagrams	4772
ARM EXPRESS listing entities	4468
mapping table	2389
contact_size_dependent_non_functional_land	
application assertion	1192
application object	312
ARM diagrams	4772
ARM EXPRESS listing entities	4468
mapping table	2390
context_dependent_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3814
context_dependent_shape_representation	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	3884
context_dependent_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3884
contract	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3884
AIM EXPRESS short listing entities	3483
application assertion	1152, 1192–1193, 1225, 1359
application object	313
ARM diagrams	4655
ARM EXPRESS listing entities	4468
mapping table	1610
contract_assigned_item	
AIM diagrams	4873
AIM EXPRESS expanded listing types	3814
AIM EXPRESS short listing types	2951
contract_assignment	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3884

contract_constraint	
AIM EXPRESS expanded listing rules	4104
contract_type	
AIM diagrams	4873
AIM EXPRESS expanded listing entities	3885
AIM EXPRESS short listing entities	3484
contract_unique_constraint	
AIM EXPRESS expanded listing rules	4104
AIM EXPRESS short listing rules	3597
conversion_based_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3885
mapping table	2837
coordinated_assembly_and_shape	
AIM EXPRESS expanded listing rules	4105
coordinated_characteristic	
application assertion	1193, 1204, 1226, 1230, 1271, 1283, 1287, 1296, 1300, 1315
application object	314
ARM diagrams	4661
ARM EXPRESS listing entities	4468
mapping table	1643
coordinated_representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3886
AIM EXPRESS short listing entities	3082
mapping table	1643, 1647, 1652–1654, 1745, 1752, 1776, 1790, 1900
coordinated_universal_time_offset	
AIM diagrams	4883
AIM EXPRESS expanded listing entities	3886
mapping table	1612–1613
correlated_or_independent	
ARM diagrams	4755
ARM EXPRESS listing types	4404
correlated_terminals	
ARM EXPRESS listing functions	4610
count_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3814
counterbore_passage_template	
application assertion	1193
application object	316
ARM diagrams	4672
ARM EXPRESS listing entities	4468
mapping table	1732, 1945, 2405
countersunk_passage_template	
application assertion	1193

application object	317
ARM diagrams	4672
ARM EXPRESS listing entities	4468
mapping table	1732, 1946, 2406
cross_product	
AIM EXPRESS expanded listing functions	4265
csg_2d_shape	
application object	318
ARM EXPRESS listing entities	4469
mapping table	2841
csg_2d_shape_representation	
AIM EXPRESS short listing entities	3085
mapping table	2841
csg_primitive	
AIM diagrams	4925
AIM EXPRESS expanded listing types	3815
application assertion	1167
application object	318
ARM diagrams	4690
ARM EXPRESS listing entities	4469
mapping table	1658
csg_select	
AIM diagrams	4926
AIM EXPRESS expanded listing types	3815
csg_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3887
mapping table	1658
csg_solid	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	3887
application object	319
ARM diagrams	4823
ARM EXPRESS listing entities	4469
mapping table	1658
curve	
AIM diagrams	4911
AIM EXPRESS expanded listing entities	3887
application assertion	1184, 1186, 1319, 1339
application object	319
ARM diagrams	4824
ARM EXPRESS listing entities	4469
mapping table	2841
curve_bounded_surface	
AIM diagrams	4909
AIM EXPRESS expanded listing entities	3887

curve_dimension	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3888
AIM EXPRESS short listing entities	3087
application object	319
ARM diagrams	4702
ARM EXPRESS listing entities	4469
mapping table	1847
curve_extension_constraint	
AIM EXPRESS short listing rules	3598
curve_font_or_scaled_curve_font_select	
AIM diagrams	4928
AIM EXPRESS expanded listing types	3815
ARM diagrams	4837
ARM EXPRESS listing types	4404
curve_on_surface	
AIM diagrams	4912
AIM EXPRESS expanded listing types	3815
curve_or_annotation_curve_occurrence	
AIM diagrams	4934
AIM EXPRESS expanded listing types	3815
curve_or_render	
AIM diagrams	4928
AIM EXPRESS expanded listing types	3815
curve_replica	
AIM diagrams	4911
AIM EXPRESS expanded listing entities	3888
curve_style	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3888
AIM EXPRESS short listing entities	3484
application assertion	1194, 1233–1234, 1339, 1347
application object	320
ARM diagrams	4837
ARM EXPRESS listing entities	4469
mapping table	2921
curve_style_font	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	3888
application assertion	1194
application object	326
ARM diagrams	4838
ARM EXPRESS listing entities	4470
mapping table	2924
curve_style_font_and_scaling	
AIM diagrams	4928

AIM EXPRESS expanded listing entities	3888
application assertion	1194
application object	327
ARM diagrams	4838
ARM EXPRESS listing entities	4470
mapping table	2924
curve_style_font_and_scaling_unique_constraint	
AIM EXPRESS expanded listing rules	4105
AIM EXPRESS short listing rules	3599
curve_style_font_pattern	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	3888
curve_style_font_select	
AIM diagrams	4932
AIM EXPRESS expanded listing types	3815
ARM diagrams	4838
ARM EXPRESS listing types	4404
curve_style_requires_length_measure_with_unit	
AIM EXPRESS expanded listing rules	4105
AIM EXPRESS short listing rules	3600
curve_style_unique_constraint	
AIM EXPRESS expanded listing rules	4106
AIM EXPRESS short listing rules	3601
curve_style_with_ends_and_corners	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3888
curve_style_with_extension	
AIM diagrams	4928
AIM EXPRESS expanded listing entities	3888
curve_weights_positive	
AIM EXPRESS expanded listing functions	4265
cutout	
application assertion	1195, 1215, 1245, 1262
application object	328
ARM diagrams	4747
ARM EXPRESS listing entities	4470
mapping table	2094
cutout_edge_segment	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3889
AIM EXPRESS short listing entities	3087
application assertion	1195
application object	330
ARM diagrams	4668
ARM EXPRESS listing entities	4470
mapping table	2095

cylindrical_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	3889
cylindricity_tolerance	
application object	331
ARM diagrams	4696
ARM EXPRESS listing entities	4470
mapping table	1847
data_element	
application assertion	1160, 1171, 1230, 1265, 1268, 1273, 1275, 1301–1302, 1326
application object	332
ARM diagrams	4659
ARM EXPRESS listing entities	4471
mapping table	1644
data_environment	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	3889
mapping table	1643
data_set_representation_item	
AIM EXPRESS short listing entities	3088
mapping table	1641, 2868
date	
AIM diagrams	4883
AIM EXPRESS expanded listing entities	3889
AIM EXPRESS short listing entities	3484
application assertion	1170, 1192, 1219, 1301, 1322, 1360–1361
application object	334
ARM diagrams	4656
ARM EXPRESS listing entities	4471
mapping table	1612
date_and_time	
AIM diagrams	4883
AIM EXPRESS expanded listing entities	3889
AIM EXPRESS short listing entities	3485
application assertion	1170, 1192, 1219–1220, 1222, 1283, 1301, 1321, 1359, 1361
application object	335
ARM diagrams	4656
ARM EXPRESS listing entities	4471
mapping table	1612
date_and_time_assigned_item	
AIM diagrams	4882
AIM EXPRESS expanded listing types	3816
AIM EXPRESS short listing types	2951
date_and_time_assignment	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3890

date_and_time_require_minute_and_second	
AIM EXPRESS expanded listing rules	4106
AIM EXPRESS short listing rules	3602
date_assigned_item	
AIM diagrams	4882
AIM EXPRESS expanded listing types	3816
AIM EXPRESS short listing types	2951
date_assignment	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3890
date_constraint	
AIM EXPRESS expanded listing rules	4106
AIM EXPRESS short listing rules	3603
date_or_date_and_time	
ARM diagrams	4655
ARM EXPRESS listing types	4404
date_role	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3890
AIM EXPRESS short listing entities	3485
date_time_role	
AIM diagrams	4882
AIM EXPRESS expanded listing entities	3890
AIM EXPRESS short listing entities	3485
date_time_select	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3816
dated_effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3890
mapping table	2725
datum	
application assertion	1197, 1200
application object	337
ARM diagrams	4673
ARM EXPRESS listing entities	4472
mapping table	1733, 1847
datum_axis	
application assertion	1195, 1199, 1304–1305
application object	338
ARM diagrams	4673
ARM EXPRESS listing entities	4472
mapping table	1848
datum_axis_and_tolerance_zone_orienting_relationship	
application assertion	1195
application object	338

ARM diagrams	4702
ARM EXPRESS listing entities	4472
mapping table	1848
datum_axis_and_tolerance_zone_orienting_relationship_constraint	
AIM EXPRESS short listing rules	3604
datum_axis_related_orientation	
application assertion	1195, 1346
application object	339
ARM diagrams	4703
ARM EXPRESS listing entities	4472
mapping table	1848
datum_axis_related_orientation_constraint	
AIM EXPRESS short listing rules	3604
datum_based_direction_vector	
application assertion	1195–1196, 1218
application object	340
ARM diagrams	4703
ARM EXPRESS listing entities	4472
mapping table	1849
datum_based_direction_vector_constraint	
AIM EXPRESS short listing rules	3605
datum_based_length_measure	
application assertion	1162, 1180, 1196
application object	342
ARM diagrams	4663
ARM EXPRESS listing entities	4472
mapping table	1659
datum_based_length_measure_constraint	
AIM EXPRESS short listing rules	3605
datum_based_vector_orientation	
application assertion	1155, 1196, 1217, 1346
application object	343
ARM diagrams	4703
ARM EXPRESS listing entities	4472
mapping table	1849
datum_based_vector_orientation_constraint	
AIM EXPRESS short listing rules	3606
datum_difference	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3890
AIM EXPRESS short listing entities	3089
application assertion	1196–1197
application object	344
ARM diagrams	4856
ARM EXPRESS listing entities	4473
mapping table	1661, 1670

datum_difference_analytical_model_port_assignment	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	3891
AIM EXPRESS short listing entities	3090
application assertion	1196
application object	345
ARM diagrams	4841
ARM EXPRESS listing entities	4473
mapping table	1671
datum_difference_based_characteristic	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3891
AIM EXPRESS short listing entities	3091
application assertion	1196
application object	346
ARM diagrams	4660
ARM EXPRESS listing entities	4473
mapping table	1662
datum_difference_based_model_parameter	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3891
AIM EXPRESS short listing entities	3092
application assertion	1197
application object	347
ARM diagrams	4618
ARM EXPRESS listing entities	4473
mapping table	1672
datum_difference_functional_unit_usage_view_terminal_assignment	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	3892
AIM EXPRESS short listing entities	3093
application assertion	1197
application object	347
ARM diagrams	4857
ARM EXPRESS listing entities	4473
mapping table	1673
datum_feature	
application assertion	1197–1198
application object	348
ARM diagrams	4704
ARM EXPRESS listing entities	4473
mapping table	1850
datum_feature_material_condition_property	
application assertion	1197, 1199
application object	349
ARM diagrams	4707

ARM EXPRESS listing entities	4474
mapping table	1851
datum_feature_material_condition_property_constraint	
AIM EXPRESS short listing rules	3608
datum_feature_usage_in_common_datum_relationship	
application assertion	1197
application object	351
ARM diagrams	4708
ARM EXPRESS listing entities	4474
mapping table	1851
datum_feature_usage_in_common_datum_relationship_constraint	
AIM EXPRESS short listing rules	3609
datum_feature_usage_in_datum_relationship	
application assertion	1197, 1318
application object	352
ARM diagrams	4708
ARM EXPRESS listing entities	4474
mapping table	1851
datum_feature_usage_in_datum_relationship_constraint	
AIM EXPRESS short listing rules	3610
datum_feature_usage_in_datum_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4107
AIM EXPRESS short listing rules	3610
datum_feature_usage_in_datum_system_relationship	
application assertion	1197–1198, 1276
application object	353
ARM diagrams	4708
ARM EXPRESS listing entities	4474
mapping table	1852
datum_feature_usage_in_datum_system_relationship_constraint	
AIM EXPRESS short listing rules	3612
datum_feature_usage_in_single_datum_relationship	
application assertion	1198
application object	354
ARM diagrams	4708
ARM EXPRESS listing entities	4474
mapping table	1852
datum_feature_usage_in_single_datum_relationship_constraint	
AIM EXPRESS short listing rules	3612
datum_plane	
application assertion	1195–1196, 1199, 1218, 1304
application object	355
ARM diagrams	4673
ARM EXPRESS listing entities	4474
mapping table	1853
datum_point	

application assertion	1198
application object	355
ARM diagrams	4673
ARM EXPRESS listing entities	4474
mapping table	1853
datum_precedence_assignment	
application assertion	1198–1199
application object	356
ARM diagrams	4709
ARM EXPRESS listing entities	4475
mapping table	1854
datum_precedence_assignment_constraint	
AIM EXPRESS short listing rules	3613
datum_precedence_assignment_unique_constraint	
AIM EXPRESS expanded listing rules	4108
AIM EXPRESS short listing rules	3614
datum_reference_frame	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3893
AIM EXPRESS short listing entities	3094
application assertion	1198–1199, 1276
application object	357
ARM diagrams	4709
ARM EXPRESS listing entities	4475
mapping table	1855
datum_system	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3893
AIM EXPRESS short listing entities	3096
application assertion	1198–1200
application object	360
ARM diagrams	4710
ARM EXPRESS listing entities	4475
mapping table	1857
datum_system_based_dimensional_location	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3893
AIM EXPRESS short listing entities	3098
mapping table	1862
datum_system_defining_conditions	
application assertion	1199, 1243
application object	362
ARM diagrams	4713
ARM EXPRESS listing entities	4475
mapping table	1858
datum_system_defining_conditions_constraint	

AIM EXPRESS short listing rules	3616
datum_system_defining_conditions_with_material_conditions	
application assertion	1199
application object	364
ARM diagrams	4713
ARM EXPRESS listing entities	4475
mapping table	1858
datum_system_defining_conditions_with_material_conditions_constraint	
AIM EXPRESS short listing rules	3617
datum_system_defining_conditions_without_material_conditions	
application assertion	1217
application object	365
ARM diagrams	4713
ARM EXPRESS listing entities	4476
mapping table	1859
datum_system_defining_conditions_without_material_conditions_constraint	
AIM EXPRESS short listing rules	3618
datum_target	
application assertion	1200
application object	365
ARM diagrams	4704
ARM EXPRESS listing entities	4476
mapping table	1859
datum_target_set	
application assertion	1200
application object	366
ARM diagrams	4704
ARM EXPRESS listing entities	4476
mapping table	1859
datum_target_usage_in_datum_target_set_relationship	
application assertion	1200
application object	367
ARM diagrams	4712
ARM EXPRESS listing entities	4476
mapping table	1859
datum_target_usage_in_datum_target_set_relationship_constraint	
AIM EXPRESS short listing rules	3618
datum_target_usage_in_datum_target_set_relationship_unique_constraint	
AIM EXPRESS short listing rules	3619
datum_usage_in_datum_system_relationship	
application assertion	1198, 1200, 1276
application object	368
ARM diagrams	4713
ARM EXPRESS listing entities	4476
mapping table	1859
datum_usage_in_datum_system_relationship_constraint	

AIM EXPRESS short listing rules	3621
datum_usage_in_datum_system_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4111
AIM EXPRESS short listing rules	3622
day_in_month_number	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3816
default_assembly_bond_shape	
application assertion	1167, 1200
application object	370
ARM diagrams	4654
ARM EXPRESS listing entities	4476
mapping table	1607
default_assembly_bond_shape_constraint	
AIM EXPRESS expanded listing rules	4111
AIM EXPRESS short listing rules	3624
default_attachment_size_and_component_termination_passage_based_land_physical_template	
application assertion	1181–1182
application object	371
ARM diagrams	4779
ARM EXPRESS listing entities	4477
mapping table	2407
default_attachment_size_and_via_based_land_physical_template	
application assertion	1354
application object	371
ARM diagrams	4778
ARM EXPRESS listing entities	4477
mapping table	2407
default_attachment_size_based_land_physical_template	
application assertion	1192, 1200
application object	371
ARM diagrams	4780
ARM EXPRESS listing entities	4477
mapping table	2407
default_component_termination_passage_based_land_physical_template	
application assertion	1182, 1200
application object	373
ARM diagrams	4779
ARM EXPRESS listing entities	4477
mapping table	2408
default_component_termination_passage_definition	
application assertion	1183, 1200–1201
application object	374
ARM diagrams	4782
ARM EXPRESS listing entities	4477
mapping table	2409

default_land_physical_template	
application assertion	1201, 1204, 1283
application object	375
ARM diagrams	4780
ARM EXPRESS listing entities	4477
mapping table	2412
default_passage_based_land_physical_template	
application assertion	1201–1202
application object	376
ARM diagrams	4778
ARM EXPRESS listing entities	4478
mapping table	2413
default_passage_based_land_physical_template_unique_constraint	
AIM EXPRESS expanded listing rules	4112
AIM EXPRESS short listing rules	3624
default_trace_template	
application assertion	1202
application object	378
ARM diagrams	4799
ARM EXPRESS listing entities	4478
mapping table	2590
default_unsupported_passage_based_land_physical_template	
application assertion	1202, 1349
application object	378
ARM diagrams	4778
ARM EXPRESS listing entities	4478
mapping table	2416
default_unsupported_passage_definition	
application assertion	1202, 1349
application object	379
ARM diagrams	4782
ARM EXPRESS listing entities	4478
mapping table	2416
default_via_based_land_physical_template	
application assertion	1202, 1355
application object	380
ARM diagrams	4778
ARM EXPRESS listing entities	4478
mapping table	2417
default_via_definition	
application assertion	1202, 1355
application object	380
ARM diagrams	4782
ARM EXPRESS listing entities	4478
mapping table	2418
defined_symbol_select	

AIM diagrams	4886
AIM EXPRESS expanded listing types	3816
definitional_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3893
degenerate_pcurve	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	3894
degenerate_toroidal_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	3894
degree_of_freedom	
ARM EXPRESS listing types	4405
degree_or_radian	
ARM diagrams	4825
ARM EXPRESS listing types	4404
delete_design_object_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	3894
AIM EXPRESS short listing entities	3098
mapping table	1693
delete_design_object_management_relationship	
application assertion	1203
application object	381
ARM diagrams	4666
ARM EXPRESS listing entities	4479
mapping table	1693
delete_design_object_management_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4115
AIM EXPRESS short listing rules	3627
delete_design_object_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	3894
AIM EXPRESS short listing entities	3099
mapping table	1693
dependent_instantiable_action_directive	
AIM EXPRESS expanded listing rules	4115
AIM EXPRESS short listing rules	3630
dependent_instantiable_address	
AIM EXPRESS expanded listing rules	4115
dependent_instantiable_approval_status	
AIM EXPRESS expanded listing rules	4115
AIM EXPRESS short listing rules	3631
dependent_instantiable_certification_type	
AIM EXPRESS expanded listing rules	4115
AIM EXPRESS short listing rules	3631

dependent_instantiable_contract_type	
AIM EXPRESS expanded listing rules	4116
AIM EXPRESS short listing rules	3632
dependent_instantiable_date	
AIM EXPRESS expanded listing rules	4116
AIM EXPRESS short listing rules	3632
dependent_instantiable_date_role	
AIM EXPRESS expanded listing rules	4116
AIM EXPRESS short listing rules	3633
dependent_instantiable_date_time_role	
AIM EXPRESS expanded listing rules	4116
AIM EXPRESS short listing rules	3633
dependent_instantiable_document_type	
AIM EXPRESS expanded listing rules	4116
dependent_instantiable_named_unit	
AIM EXPRESS expanded listing rules	4116
dependent_instantiable_organization_role	
AIM EXPRESS expanded listing rules	4117
dependent_instantiable_parametric_representation_context	
AIM EXPRESS expanded listing rules	4117
dependent_instantiable_person_and_organization_role	
AIM EXPRESS expanded listing rules	4117
dependent_instantiable_representation_item	
AIM EXPRESS expanded listing rules	4117
dependent_instantiable_security_classification_level	
AIM EXPRESS expanded listing rules	4117
dependent_instantiable_shape_representation	
AIM EXPRESS expanded listing rules	4118
dependent_instantiable_tolerance_value	
AIM EXPRESS expanded listing rules	4118
dependent_material_removal_feature_template	
AIM EXPRESS short listing entities	3100
application assertion	1204, 1282
application object	382
ARM EXPRESS listing entities	4479
mapping table	2419
dependently_located_layer_connection_point	
application assertion	1203
application object	383
ARM diagrams	4735
ARM EXPRESS listing entities	4479
mapping table	2016
dependently_located_layer_connection_point_unique_constraint	
AIM EXPRESS expanded listing rules	4119
derive_dimensional_exponents	
AIM EXPRESS expanded listing functions	4266

derive_mirror	
ARM EXPRESS listing functions	4611
derived_shape_aspect	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3895
AIM EXPRESS short listing entities	3486
mapping table	1860, 1869
derived_shape_element	
application assertion	1323
application object	384
ARM diagrams	4692
ARM EXPRESS listing entities	4479
mapping table	1860
derived_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3895
mapping table	1391, 1668
derived_unit_element	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3895
descriptive_measure	
AIM diagrams	4927
AIM EXPRESS expanded listing types	3816
descriptive_or_geometric_orientation	
ARM diagrams	4776
ARM EXPRESS listing types	4405
descriptive_representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3895
AIM EXPRESS short listing entities	3486
mapping table	1389–1390, 1393–1394, 1446, 1486, 1522–1523, 1551, 1579, 1581, 1648, 1661, 1666–1667, 1670, 1673–1674, 1702, 1714, 1720, 1730, 1743–1744, 1750–1751, 1775–1776, 1779–1780, 1789–1790, 1807–1808, 1880, 1916–1917, 1919–1920, 1925, 2008, 2046, 2060, 2062–2063, 2069, 2076–2077, 2100, 2111, 2136, 2192, 2212, 2217, 2259–2261, 2266, 2329, 2421, 2423, 2430, 2435–2436, 2459, 2540, 2628–2631, 2636–2637, 2640, 2644–2645, 2655–2656, 2658–2659, 2700–2701, 2705, 2710–2711, 2715–2716, 2799, 2883–2884, 2904, 2910, 2922
design_characteristic_occurrence	
application object	385
ARM diagrams	4820
ARM EXPRESS listing entities	4479
mapping table	2804
design_composition_path	
application assertion	1151, 1204–1205
application object	386
ARM diagrams	4676
ARM EXPRESS listing entities	4479

mapping table	1742, 1749, 1773, 1787
design_composition_path_or_root	
ARM EXPRESS listing types	4405
design_functional_unit_allocation_to_assembly_component	
application assertion	1181, 1204
application object	387
ARM diagrams	4675
ARM EXPRESS listing entities	4480
mapping table	1743, 1750, 1775, 1789
design_functional_unit_allocation_to_assembly_joint	
application assertion	1204–1205
application object	389
ARM diagrams	4676
ARM EXPRESS listing entities	4480
mapping table	1746
design_functional_unit_allocation_to_fabrication_joint	
application assertion	1205
application object	390
ARM diagrams	4678
ARM EXPRESS listing entities	4480
mapping table	1753
design_functional_unit_allocation_to_reference_functional_unit	
application assertion	1205, 1241
application object	391
ARM diagrams	4684
ARM EXPRESS listing entities	4480
mapping table	1777, 1792
design_intent_modification	
application assertion	1205–1210
application object	392
ARM diagrams	4836
ARM EXPRESS listing entities	4480
mapping table	2890
design_intent_modification_3d_position	
application assertion	1207–1208
application object	396
ARM EXPRESS listing entities	4481
mapping table	2347, 2898
design_intent_modification_3d_shape	
application assertion	1207–1208
application object	397
ARM diagrams	4765
ARM EXPRESS listing entities	4481
mapping table	2348
design_intent_modification_3d_shape_select	
ARM diagrams	4766

ARM EXPRESS listing types	4405
design_intent_modification_in_stratum_3d_position	
ARM diagrams	4766
design_intent_modification_in_stratum_3d_position_constraint	
AIM EXPRESS expanded listing rules	4122
design_intent_modification_in_stratum_planar_position	
ARM diagrams	4766
design_intent_modification_in_stratum_planar_position_constraint	
AIM EXPRESS expanded listing rules	4122
design_intent_modification_non_planar_2d_shape	
application assertion	1207–1208
application object	397
ARM diagrams	4767
ARM EXPRESS listing entities	4481
mapping table	2349
design_intent_modification_or_material_removal_component	
ARM EXPRESS listing types	4405
design_intent_modification_planar_position	
application assertion	1208–1209
application object	398
ARM EXPRESS listing entities	4481
mapping table	2349, 2900
design_intent_modification_planar_shape	
application assertion	1208–1210
application object	399
ARM diagrams	4767
ARM EXPRESS listing entities	4481
mapping table	2351
design_intent_modification_planar_shape_select	
ARM diagrams	4766
ARM EXPRESS listing types	4405
design_intent_modification_sequence	
application assertion	1209
application object	400
ARM diagrams	4834
ARM EXPRESS listing entities	4481
mapping table	2901
design_intent_modification_sequence_constraint	
AIM EXPRESS expanded listing rules	4123
design_intent_modification_sequence_unique_constraint	
AIM EXPRESS expanded listing rules	4124
design_intent_modification_styled_area_representation	
application assertion	1208–1210
application object	401
ARM diagrams	4768
ARM EXPRESS listing entities	4481

mapping table	2351
design_layer_stratum	
application assertion	1188, 1190, 1210, 1258
application object	402
ARM diagrams	4748
ARM EXPRESS listing entities	4482
mapping table	2096, 2877
design_layer_technology	
application assertion	1201, 1210
application object	404
ARM diagrams	4784
ARM EXPRESS listing entities	4482
mapping table	2420
design_layer_type_specific_padstack_definition	
AIM EXPRESS short listing entities	3101
ARM EXPRESS listing entities	4482
mapping table	2590
design_make_from_relationship	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	3895
AIM EXPRESS short listing entities	3101
mapping table	1380
design_object	
AIM diagrams	4863
AIM EXPRESS expanded listing entities	3896
AIM EXPRESS short listing entities	3102
design_object_management_relationship	
application assertion	1171, 1210
application object	405
ARM diagrams	4666
ARM EXPRESS listing entities	4482
mapping table	1694
design_object_management_relationship_or_description	
ARM diagrams	4664
ARM EXPRESS listing types	4405
design_requirement_occurrence	
application object	406
ARM diagrams	4821
ARM EXPRESS listing entities	4482
mapping table	2804
design_specification	
application object	407
ARM diagrams	4667
ARM EXPRESS listing entities	4482
mapping table	1708
design_view_connection_zone_position_relationship	

application object	407
ARM EXPRESS listing entities	4483
mapping table	1594, 2353
design_view_connection_zone_terminal_shape_relationship	
application assertion	1211–1212
design_view_terminal_component_shape_relationship	
application assertion	1212–1213
application object	409
ARM EXPRESS listing entities	4483
mapping table	1597, 2357
device_2d_position	
application assertion	1213–1214
application object	410
ARM diagrams	4796
ARM EXPRESS listing entities	4483
mapping table	2515
device_3d_position	
application assertion	1214
application object	411
ARM diagrams	4668
ARM EXPRESS listing entities	4483
mapping table	2518
device_terminal_map	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3896
AIM EXPRESS short listing entities	3103
application assertion	1214
application object	412
ARM diagrams	4796
ARM EXPRESS listing entities	4483
mapping table	2522
device_terminal_map_relationship	
application assertion	1214
application object	413
ARM diagrams	4796
ARM EXPRESS listing entities	4483
mapping table	2522
device_terminal_map_relationship_constraint	
AIM EXPRESS expanded listing rules	4124
diametrical_boundary_size_characteristic	
application assertion	1215
application object	414
ARM diagrams	4694
ARM EXPRESS listing entities	4484
mapping table	1860
dielectric_crossover_area	

application assertion	1215
application object	415
ARM diagrams	4736
ARM EXPRESS listing entities	4484
mapping table	2018
dielectric_material_passage	
application assertion	1215
application object	416
ARM diagrams	4749
ARM EXPRESS listing entities	4484
mapping table	2098
digital_analytical_model_port	
application object	418
ARM diagrams	4617
ARM EXPRESS listing entities	4484
mapping table	1393, 1673
digital_signal_flow_direction	
ARM diagrams	4617
dimension_along_curve	
application assertion	1216
application object	418
ARM diagrams	4714
ARM EXPRESS listing entities	4484
mapping table	1860
dimension_along_curve_unique_constraint	
AIM EXPRESS expanded listing rules	4124
dimension_count	
AIM diagrams	4902
AIM EXPRESS expanded listing types	3816
dimension_location_with_direction_vector	
application assertion	1216–1217
application object	419
ARM diagrams	4714
ARM EXPRESS listing entities	4484
mapping table	1861
dimension_location_with_direction_vector_unique_constraint	
AIM EXPRESS expanded listing rules	4126
dimension_location_with_specified_datum_system	
application assertion	1217
application object	420
ARM diagrams	4714
ARM EXPRESS listing entities	4485
mapping table	1862
dimension_of	
AIM EXPRESS expanded listing functions	4267
dimension_qualifier	

application assertion	1217
application object	421
ARM diagrams	4715
ARM EXPRESS listing entities	4485
mapping table	1863
dimension_qualifiers	
ARM diagrams	4715
ARM EXPRESS listing types	4406
dimension_related_positional_boundary	
application assertion	1217–1218
application object	422
ARM diagrams	4712
ARM EXPRESS listing entities	4485
mapping table	1864
dimension_related_positional_boundary_definition	
application assertion	1217–1218
application object	423
ARM diagrams	4716
ARM EXPRESS listing entities	4485
mapping table	1864
dimension_related_tolerance_zone_element	
AIM diagrams	4935
AIM EXPRESS expanded listing entities	3896
dimension_set	
application assertion	1218, 1348
application object	424
ARM diagrams	4715
ARM EXPRESS listing entities	4485
mapping table	1865
dimensional_characteristic	
AIM diagrams	4898
AIM EXPRESS expanded listing types	3817
application assertion	1217–1218, 1329, 1344
application object	425
ARM diagrams	4717
ARM EXPRESS listing entities	4485
mapping table	1866
dimensional_characteristic_representation	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3896
application assertion	1215–1216
application object	426
ARM diagrams	4716
ARM EXPRESS listing entities	4485
mapping table	1867
dimensional_exponents	

AIM diagrams	4894
AIM EXPRESS expanded listing entities	3896
dimensional_location	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3896
AIM EXPRESS short listing entities	3487
application assertion	1215–1216
application object	427
ARM diagrams	4714
ARM EXPRESS listing entities	4486
mapping table	1866, 1868, 1919
dimensional_location_or_dimensional_size	
ARM diagrams	4716
ARM EXPRESS listing types	4406
dimensional_location_with_direction	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3897
AIM EXPRESS short listing entities	3103
mapping table	1861
dimensional_location_with_path	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3897
mapping table	1860
dimensional_size	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3897
AIM EXPRESS short listing entities	3487
application assertion	1215–1216, 1303
application object	428
ARM diagrams	4702
ARM EXPRESS listing entities	4486
mapping table	1866, 1868, 1921
dimensional_size_based_on_opposing_boundaries	
application assertion	1216
application object	429
ARM diagrams	4702
ARM EXPRESS listing entities	4486
mapping table	1869
dimensional_size_property	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3897
AIM EXPRESS short listing entities	3104
dimensional_types	
ARM diagrams	4717
ARM EXPRESS listing types	4406
dimensions_for_si_unit	

AIM EXPRESS expanded listing functions	4267
directed_action	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3897
AIM EXPRESS short listing entities	3488
mapping table	1679–1680
directed_action_requires_approval	
AIM EXPRESS expanded listing rules	4126
directed_action_requires_date_or_date_and_time	
AIM EXPRESS expanded listing rules	4126
directed_axis	
application assertion	1218, 1321
application object	431
ARM diagrams	4711
ARM EXPRESS listing entities	4486
mapping table	1869
directed_axis_constraint	
AIM EXPRESS expanded listing rules	4127
directed_axis_unique_constraint	
AIM EXPRESS expanded listing rules	4128
directed_dimensional_location	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	3897
AIM EXPRESS short listing entities	3105
mapping table	1868
direction	
AIM diagrams	4906
AIM EXPRESS expanded listing entities	3897
application object	432
ARM EXPRESS listing entities	4486
mapping table	2841
direction_element	
application assertion	1196, 1218
application object	432
ARM diagrams	4718
ARM EXPRESS listing entities	4486
mapping table	1870
direction_element_constraint	
AIM EXPRESS expanded listing rules	4129
discrete_shield	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3898
AIM EXPRESS short listing entities	3105
mapping table	1426
disjoint_package_terminals	
ARM EXPRESS listing functions	4611

document	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
AIM EXPRESS short listing entities	3488
mapping table	1708, 1721, 2764
document_assigned_item	
AIM diagrams	4876
AIM EXPRESS expanded listing types	3817
AIM EXPRESS short listing types	2952
document_identifier	
AIM EXPRESS short listing entities	3106
mapping table	1713
document_identifier_assigned_item	
AIM EXPRESS short listing types	2953
document_identifier_assignment	
AIM EXPRESS short listing entities	3107
document_or_organization	
ARM diagrams	4817
ARM EXPRESS listing types	4406
document_reference	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
document_relationship	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
document_type	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
AIM EXPRESS short listing entities	3489
document_usage_constraint	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
mapping table	1720
document_with_class	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3898
documentation_layer_stratum	
application assertion	1219
application object	434
ARM diagrams	4748
ARM EXPRESS listing entities	4487
mapping table	2099, 2878
documentation_layer_technology	
application assertion	1219
application object	436
ARM diagrams	4784

ARM EXPRESS listing entities	4487
mapping table	2423
dose_equivalent_data_element	
application object	436
ARM EXPRESS listing entities	4487
mapping table	1645
dose_equivalent_measure_with_unit	
AIM EXPRESS short listing entities	3107
mapping table	1645
dot_product	
AIM EXPRESS expanded listing functions	4268
double-sided printed circuit board	9
edge	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	3899
edge_based_2d_wireframe	
application object	437
ARM diagrams	4827
ARM EXPRESS listing entities	4487
mapping table	2931
edge_based_2d_wireframe_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3900
AIM EXPRESS short listing entities	3108
mapping table	2931
edge_based_wireframe	
application object	438
ARM diagrams	4823
ARM EXPRESS listing entities	4487
mapping table	2932
edge_based_wireframe_model	
AIM diagrams	4918
AIM EXPRESS expanded listing entities	3901
edge_based_wireframe_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3902
mapping table	2932
edge_curve	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	3902
edge_loop	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3902
edge_reversed	
AIM EXPRESS expanded listing functions	4268
edge_segment_cross_section	

AIM EXPRESS short listing entities	3111
application assertion	1219, 1252
application object	438
ARM EXPRESS listing entities	4487
mapping table	1733
edge_segment_vertex	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3903
AIM EXPRESS short listing entities	3112
application assertion	1165, 1195, 1246–1247, 1252, 1280, 1308
application object	439
ARM diagrams	4749
ARM EXPRESS listing entities	4487
mapping table	2100, 2212, 2217, 2540
edge_segment_vertex_constraint	
AIM EXPRESS expanded listing rules	4130
edge_segment_vertex_unique_constraint	
AIM EXPRESS expanded listing rules	4131
edge_shape	
application assertion	1219
application object	440
ARM diagrams	4674
ARM EXPRESS listing entities	4488
mapping table	1735
ee_approval	
application assertion ...	1170, 1188, 1192, 1219–1220, 1222–1223, 1225, 1236, 1301, 1322, 1340, 1360, 1362
application object	440
ARM diagrams	4765
ARM EXPRESS listing entities	4488
mapping table	1613
ee_colour	
application assertion	1190
application object	442
ARM diagrams	4662
ARM EXPRESS listing entities	4488
mapping table	1645
ee_colour_unique_constraint	
AIM EXPRESS expanded listing rules	4131
ee_document	
application assertion	1153, 1171, 1193, 1220–1222, 1227–1228, 1231, 1271, 1315, 1348
application object	442
ARM diagrams	4669
ARM EXPRESS listing entities	4488
mapping table	1708
ee_document_constraint	

AIM EXPRESS expanded listing rules	4132
ee_document_identifier	
application assertion	1220
application object	445
ARM diagrams	4667
ARM EXPRESS listing entities	4489
mapping table	1713
ee_document_identifier_unique_constraint	
AIM EXPRESS expanded listing rules	4132
ee_document_unique_constraint	
AIM EXPRESS expanded listing rules	4132
ee_material	
application assertion 1155, 1157, 1160, 1165, 1201–1202, 1215, 1221, 1225, 1267–1268, 1279–1280, 1284, 1292, 1338, 1340	
application object	446
ARM diagrams	4787
ARM EXPRESS listing entities	4489
mapping table	2443
ee_material_unique_constraint	
AIM EXPRESS expanded listing rules	4133
ee_name	
ARM diagrams	4616
ARM EXPRESS listing types	4406
ee_product	
application assertion 1152, 1221, 1224–1226, 1316, 1327, 1339	
application object	447
ARM diagrams	4813
ARM EXPRESS listing entities	4489
mapping table	2760, 2792
ee_product_category	
application assertion	1221–1222
application object	450
ARM diagrams	4788
ARM EXPRESS listing entities	4490
mapping table	2444, 2666, 2761
ee_product_category_relationship	
application assertion	1222
application object	451
ARM diagrams	4787
ARM EXPRESS listing entities	4490
mapping table	2446, 2668, 2763
ee_product_configuration	
application assertion 1222, 1301, 1316, 1326	
application object	452
ARM diagrams	4670
ARM EXPRESS listing entities	4490

mapping table	1718
ee_product_configuration_unique_constraint	
AIM EXPRESS expanded listing rules	4134
ee_product_constraint	
AIM EXPRESS expanded listing rules	4135
ee_product_definition	
application assertion	1210, 1222–1223, 1230, 1316, 1326
application object	454
ARM diagrams	4814
ARM EXPRESS listing entities	4491
mapping table	2763
ee_product_definition_based_constraint_occurrence	
application assertion	1223
application object	457
ARM EXPRESS listing entities	4491
mapping table	2804
ee_product_definition_constraint	
AIM EXPRESS expanded listing rules	4135
ee_product_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4137
ee_product_definition_with_annotation_elements	
ARM EXPRESS listing types	4407
ee_product_model	
application assertion	1222–1223
application object	458
ARM diagrams	4670
ARM EXPRESS listing entities	4491
mapping table	1719
ee_product_model_unique_constraint	
AIM EXPRESS expanded listing rules	4137
ee_product_related_product_category	
application assertion	1224
application object	459
ARM diagrams	4788
ARM EXPRESS listing entities	4491
mapping table	2447, 2668, 2782
ee_product_specific_parameter_value_assignment	
application assertion	1224
application object	459
ARM diagrams	4802
ARM EXPRESS listing entities	4491
mapping table	2668, 2782
ee_product_unique_constraint	
AIM EXPRESS expanded listing rules	4139
ee_product_version	
application assertion	1188, 1222–1225, 1273, 1312, 1316, 1326, 1340, 1343, 1360–1361

application object	461
ARM diagrams	4815
ARM EXPRESS listing entities	4491
mapping table	2784
ee_product_version_constraint	
AIM EXPRESS expanded listing rules	4139
ee_product_version_supersedence	
application assertion	1225
application object	462
ARM EXPRESS listing entities	4492
mapping table	2786
ee_product_version_unique_constraint	
AIM EXPRESS expanded listing rules	4139
ee_requirement_occurrence	
application assertion ... 1162, 1176, 1206, 1215, 1225–1226, 1228, 1239, 1244, 1254, 1270, 1275, 1288, 1291, 1294, 1296–1300, 1316, 1318–1319, 1328–1330, 1342–1343	
application object	463
ARM diagrams	4820
ARM EXPRESS listing entities	4492
mapping table	2806
ee_rule	
application assertion	1228
ARM diagrams	4789
ee_rule_complex_clause	
application assertion	1226–1227
application object	467
ARM diagrams	4830
ARM EXPRESS listing entities	4492
mapping table	2793
ee_rule_conclusion_definition	
application assertion	1227
application object	468
ARM diagrams	4803
ARM EXPRESS listing entities	4493
mapping table	2793
ee_rule_conclusion_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4140
ee_rule_definition	
application assertion	1227
application object	470
ARM diagrams	4816
ARM EXPRESS listing entities	4493
mapping table	2794
ee_rule_definition_constraint	
AIM EXPRESS expanded listing rules	4141
ee_rule_definition_unique_constraint	

AIM EXPRESS expanded listing rules	4142
ee_rule_function	
application assertion	1227–1228, 1348
application object	470
ARM diagrams	4816
ARM EXPRESS listing entities	4493
mapping table	2794
ee_rule_function_unique_constraint	
AIM EXPRESS expanded listing rules	4143
ee_rule_premise_definition	
application assertion	1227
application object	472
ARM diagrams	4803
ARM EXPRESS listing entities	4493
mapping table	2795
ee_rule_premise_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4144
ee_rule_simple_clause	
application assertion	1226–1228
application object	473
ARM diagrams	4830
ARM EXPRESS listing entities	4494
mapping table	2796
ee_rule_unique_constraint	
AIM EXPRESS expanded listing rules	4145
ee_rule_version	
application assertion	1227–1228, 1319–1320
application object	474
ARM diagrams	4817
ARM EXPRESS listing entities	4494
mapping table	2796
ee_specification	
AIM diagrams	4875
AIM EXPRESS expanded listing entities	3903
AIM EXPRESS short listing entities	3112
application assertion	1153, 1226–1228, 1266, 1269, 1311, 1329
application object	475
ARM diagrams	4633
ARM EXPRESS listing entities	4494
mapping table	1394, 1483, 1487, 1607, 1674, 1703, 1708, 1713–1717, 2005, 2009, 2423
ee_text	
application assertion ...	1170–1171, 1192, 1219–1221, 1223, 1225, 1268–1269, 1338, 1360–1361
application object	475
ARM diagrams	4667
ARM EXPRESS listing entities	4494
mapping table	1714

effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3903
electric_charge_data_element	
application object	476
ARM EXPRESS listing entities	4494
mapping table	1646
electric_charge_measure_with_unit	
AIM EXPRESS short listing entities	3114
mapping table	1646
electric_current_data_element	
application object	477
ARM EXPRESS listing entities	4494
mapping table	1646
electric_current_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3817
electric_current_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3903
mapping table	1646
electric_current_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3904
electrical_isolation_component_shape_aspect	
AIM EXPRESS short listing entities	3115
mapping table	2101
electrical_isolation_laminate_component	
application assertion	1228
application object	477
ARM EXPRESS listing entities	4494
mapping table	2101
electrical_isolation_removal_structure	
application assertion	1228
application object	478
ARM diagrams	4836
ARM EXPRESS listing entities	4494
mapping table	2902
electrical_isolation_removal_template	
application assertion	1228–1229
application object	478
ARM EXPRESS listing entities	4495
mapping table	2590
electrical_isolation_removal_template_definition	
AIM EXPRESS short listing entities	3115
mapping table	2590

electrical_network	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3904
AIM EXPRESS short listing entities	3116
mapping table	1759
electrical_network_definition	
application object	481
ARM diagrams	4681
ARM EXPRESS listing entities	4495
mapping table	1759
electrical_requirement_occurrence	
application object	481
ARM diagrams	4821
ARM EXPRESS listing entities	4495
mapping table	2811
electromagnetic_compatibility_requirement_allocation	
application object	482
ARM diagrams	4647
ARM EXPRESS listing entities	4495
mapping table	1562, 2239
electromagnetic_requirement_occurrence	
application assertion	1324
application object	482
ARM diagrams	4821
ARM EXPRESS listing entities	4495
mapping table	2811
electromotive_force_data_element	
application object	482
ARM EXPRESS listing entities	4495
mapping table	1646
electromotive_force_measure_with_unit	
AIM EXPRESS short listing entities	3117
mapping table	1646
elementary_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	3904
ellipse	
AIM diagrams	4914
AIM EXPRESS expanded listing entities	3904
elliptic_area	
application assertion	1229
application object	483
ARM EXPRESS listing entities	4495
mapping table	2841
embedded_component_terminal	
application assertion	1203, 1229, 1232, 1260

application object	484
ARM EXPRESS listing entities	4495
mapping table	2101
energy_data_element	
application object	485
ARM EXPRESS listing entities	4495
mapping table	1646
energy_measure_with_unit	
AIM EXPRESS short listing entities	3117
mapping table	1646
engineering_make_from	
application assertion	1230, 1240, 1285
application object	486
ARM diagrams	4622
ARM EXPRESS listing entities	4496
mapping table	1380
environment	
ARM diagrams	4801
ARM EXPRESS listing types	4407
equivalent_functional_terminals_assignment	
application assertion	1230
application object	487
ARM diagrams	4680
ARM EXPRESS listing entities	4496
mapping table	1827
equivalent_functional_unit_definition_assignment	
application assertion	1230
application object	488
ARM diagrams	4688
ARM EXPRESS listing entities	4496
mapping table	1818
evaluated_characteristic	
AIM EXPRESS short listing entities	3118
application assertion	1230, 1361
application object	489
ARM diagrams	4662
ARM EXPRESS listing entities	4496
mapping table	2720, 2723
evaluated_characteristic_or_description	
ARM diagrams	4665
ARM EXPRESS listing types	4407
evaluated_degenerate_pcurve	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	3904
evaluation_ee_product_definition	
application object	489

ARM EXPRESS listing entities	4496
mapping table	2721
evaluation_product_definition	
AIM EXPRESS short listing entities	3119
mapping table	2721, 2763
evaluation_result_value	
ARM EXPRESS listing types	4407
evaluation_type	
ARM diagrams	4662
ARM EXPRESS listing types	4407
event_data_element	
application object	490
ARM EXPRESS listing entities	4496
mapping table	1646
executed_action	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	3904
expanded_uncertainty	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	3905
application object	491
ARM EXPRESS listing entities	4496
mapping table	1646
explicitly_located_layer_connection_point	
application assertion	1255
application object	491
ARM diagrams	4735
ARM EXPRESS listing entities	4497
mapping table	2019
explicitly_located_layer_connection_point_unique_constraint	
AIM EXPRESS expanded listing rules	4146
extend_or_chord_2_extend_or_truncate_or_round	
ARM diagrams	4837
ARM EXPRESS listing types	4407
extend_or_truncate	
ARM diagrams	4837
ARM EXPRESS listing types	4408
extension	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3905
mapping table	1883
external_definition	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3905
AIM EXPRESS short listing entities	3120

application assertion	1155, 1168, 1171, 1185, 1230–1231, 1236, 1256–1257, 1288, 1290, 1297–1299, 1324, 1330, 1339–1340
application object	493
ARM diagrams	4671
ARM EXPRESS listing entities	4497
mapping table	1720
external_source	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3905
mapping table	2786
external_source_relationship	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3905
externally_defined_assembly_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3905
AIM EXPRESS short listing entities	3120
mapping table	1550
externally_defined_bare_die	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3906
AIM EXPRESS short listing entities	3121
mapping table	1618
externally_defined_curve_font	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3906
application assertion	1194
application object	494
ARM diagrams	4838
ARM EXPRESS listing entities	4497
mapping table	2925
externally_defined_dimension_representation	
application assertion	1230
application object	494
ARM diagrams	4719
ARM EXPRESS listing entities	4497
mapping table	1872
externally_defined_ee_product_definition	
application assertion	1156
application object	495
ARM diagrams	4814
ARM EXPRESS listing entities	4497
mapping table	2786
externally_defined_functional_unit	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3906

AIM EXPRESS short listing entities	3123
mapping table	1762, 1769, 1828, 2763, 2786
externally_defined_hatch_style	
AIM diagrams	4902
AIM EXPRESS expanded listing entities	3906
application assertion	1234
application object	496
ARM diagrams	4828
ARM EXPRESS listing entities	4497
mapping table	2841
externally_defined_interconnect_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3907
AIM EXPRESS short listing entities	3123
mapping table	2234
externally_defined_item	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3907
AIM EXPRESS short listing entities	3489
mapping table	2926
externally_defined_package	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3908
AIM EXPRESS short listing entities	3124
mapping table	1366, 1372, 2450, 2458
externally_defined_packaged_connector	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3908
AIM EXPRESS short listing entities	3126
mapping table	1548, 2498
externally_defined_packaged_part	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3909
AIM EXPRESS short listing entities	3126
mapping table	1377, 2496, 2500, 2524
externally_defined_physical_unit	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3909
AIM EXPRESS short listing entities	3129
mapping table	1542, 2225, 2670, 2677, 2763, 2786
externally_defined_product_definition	
AIM EXPRESS short listing entities	3129
mapping table	2763, 2786
externally_defined_representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3909

AIM EXPRESS short listing entities	3130
externally_defined_symbol	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3910
externally_defined_text_font	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3910
AIM EXPRESS short listing entities	3489
mapping table	2926
externally_defined_tile	
AIM diagrams	4886
AIM EXPRESS expanded listing entities	3910
application assertion	1235
application object	496
ARM diagrams	4828
ARM EXPRESS listing entities	4497
mapping table	2841
externally_defined_tile_style	
AIM diagrams	4902
AIM EXPRESS expanded listing entities	3910
application assertion	1234
application object	496
ARM diagrams	4828
ARM EXPRESS listing entities	4497
mapping table	2841
extruded_area_solid	
AIM diagrams	4925
AIM EXPRESS expanded listing entities	3910
mapping table	2872
extruded_face_solid	
AIM diagrams	4935
AIM EXPRESS expanded listing entities	3910
fabricated_feature_select	
ARM diagrams	4671
ARM EXPRESS listing types	4408
fabrication_joint	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3911
AIM EXPRESS short listing entities	3130
application assertion	1172, 1205, 1231–1232
application object	497
ARM diagrams	4671
ARM EXPRESS listing entities	4498
mapping table	1723, 2019, 2102
fabrication_technology_specification	
application object	500

ARM diagrams	4670
ARM EXPRESS listing entities	4498
mapping table	1714, 2423
fabrication_thickness_requirement_occurrence	
application object	500
ARM EXPRESS listing entities	4498
mapping table	2811
face	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3911
face_bound	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	3911
face_bound_reversed	
AIM EXPRESS expanded listing functions	4269
face_outer_bound	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	3911
face_reversed	
AIM EXPRESS expanded listing functions	4269
face_surface	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3911
faceted_brep	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	3911
faceted_brep_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3913
feature_group_relationship	
application assertion	1232
application object	501
ARM diagrams	4797
ARM EXPRESS listing entities	4498
mapping table	2540
feature_group_relationship_constraint	
AIM EXPRESS expanded listing rules	4146
feature_shape_composition_relationship	
ARM diagrams	4828
feature_shape_definition	
application assertion	1211–1212, 1267, 1273, 1330, 1354
application object	502
ARM diagrams	4827
ARM EXPRESS listing entities	4498
mapping table	2841
feature_shape_occurrence	

ARM diagrams	4826
feature_shape_occurrence_constraint	
AIM EXPRESS expanded listing rules	4147
feature_shape_occurrence_relationship	
AIM EXPRESS short listing entities	3132
fiducial	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3913
AIM EXPRESS short listing entities	3133
application object	502
ARM diagrams	4744
ARM EXPRESS listing entities	4498
mapping table	2106
fiducial_part_feature	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3913
AIM EXPRESS short listing entities	3133
application object	503
ARM diagrams	4792
ARM EXPRESS listing entities	4498
mapping table	2457, 2542
fiducial_stratum_feature	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3913
AIM EXPRESS short listing entities	3134
application object	503
ARM diagrams	4707
ARM EXPRESS listing entities	4498
mapping table	2107
fill_area	
application assertion	1233
application object	504
ARM diagrams	4828
ARM EXPRESS listing entities	4499
mapping table	2842
fill_area_occurrence	
application assertion	1209, 1233, 1334
application object	504
ARM diagrams	4828
ARM EXPRESS listing entities	4499
mapping table	2842
fill_area_occurrence_constraint	
AIM EXPRESS expanded listing rules	4147
fill_area_style	
AIM diagrams	4930
AIM EXPRESS expanded listing entities	3913

application assertion	1233–1234
application object	506
ARM diagrams	4829
ARM EXPRESS listing entities	4499
mapping table	2844
fill_area_style_colour	
AIM diagrams	4930
AIM EXPRESS expanded listing entities	3914
mapping table	2857
fill_area_style_constraint	
AIM EXPRESS expanded listing rules	4147
fill_area_style_hatching	
AIM diagrams	4930
AIM EXPRESS expanded listing entities	3914
application assertion	1234
application object	507
ARM diagrams	4831
ARM EXPRESS listing entities	4499
mapping table	2845
fill_area_style_tile	
application assertion	1234–1235
application object	508
ARM diagrams	4831
ARM EXPRESS listing entities	4499
mapping table	2846
fill_area_style_tile_coloured_region	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3914
mapping table	2847
fill_area_style_tile_curve_with_style	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3914
application assertion	1235
application object	509
ARM diagrams	4832
ARM EXPRESS listing entities	4499
mapping table	2847
fill_area_style_tile_shape_select	
AIM diagrams	4934
AIM EXPRESS expanded listing types	3817
ARM diagrams	4831
ARM EXPRESS listing types	4408
fill_area_style_tile_solid_region	
application assertion	1235
application object	510
ARM diagrams	4832

ARM EXPRESS listing entities	4499
mapping table	2847
fill_area_style_tile_symbol_with_style	
AIM diagrams	4934
AIM EXPRESS expanded listing entities	3914
application assertion	1235
application object	510
ARM diagrams	4832
ARM EXPRESS listing entities	4500
mapping table	2847
fill_area_style_tiles	
AIM diagrams	4930
AIM EXPRESS expanded listing entities	3914
mapping table	2846
fill_area_style_unique_constraint	
AIM EXPRESS expanded listing rules	4148
fill_style_select	
AIM diagrams	4930
AIM EXPRESS expanded listing types	3818
ARM diagrams	4829
ARM EXPRESS listing types	4408
filled_area_material_removal_component_shape_aspect	
AIM EXPRESS short listing entities	3134
mapping table	2107
filled_area_material_removal_laminate_component	
application assertion	1233
application object	510
ARM EXPRESS listing entities	4500
mapping table	2107
filled_area_material_removal_structure	
application assertion	1233
application object	511
ARM diagrams	4836
ARM EXPRESS listing entities	4500
mapping table	2902
first_proj_axis	
AIM EXPRESS expanded listing functions	4270
flatness_tolerance	
application object	512
ARM diagrams	4696
ARM EXPRESS listing entities	4500
mapping table	1873
flexible printed circuit board	10
font_definition	
application assertion	1292, 1339
application object	513

ARM diagrams	4799
ARM EXPRESS listing entities	4500
mapping table	2926
font_select	
AIM diagrams	4933
AIM EXPRESS expanded listing types	3818
ARM diagrams	4799
ARM EXPRESS listing types	4408
footprint_definition	
AIM EXPRESS short listing entities	3134
application assertion	1235–1236, 1274
application object	513
ARM EXPRESS listing entities	4500
mapping table	2591, 2763
footprint_definition_planar_shape	
application object	514
footprint_definition_shape	
ARM EXPRESS listing entities	4500
mapping table	2628
footprint_occurrence	
AIM EXPRESS short listing entities	3136
application object	515
ARM EXPRESS listing entities	4501
mapping table	2108
footprint_occurrence_shape_aspect_relationship	
AIM EXPRESS short listing entities	3136
mapping table	2108
footprint_occurrence_sub_assembly_relationship	
application object	517
ARM EXPRESS listing entities	4501
mapping table	2108
force_data_element	
application object	518
ARM EXPRESS listing entities	4501
mapping table	1646
force_measure_with_unit	
AIM EXPRESS short listing entities	3137
mapping table	1646
founded_item	
AIM diagrams	4937
AIM EXPRESS expanded listing entities	3914
founded_item_select	
AIM diagrams	4936
AIM EXPRESS expanded listing types	3818
frequency_data_element	
application object	519

ARM EXPRESS listing entities	4501
mapping table	1646
frequency_measure_with_unit	
AIM EXPRESS short listing entities	3138
mapping table	1646
frozen_assigned_item	
AIM EXPRESS short listing types	2954
mapping table	1695
frozen_assignment	
AIM EXPRESS short listing entities	3138
application assertion	1236
application object	519
ARM diagrams	4750
ARM EXPRESS listing entities	4501
mapping table	2109
frozen_design_object	
application assertion	1236
application object	520
ARM diagrams	4667
ARM EXPRESS listing entities	4501
mapping table	1695
functional_object	10
functional_specification	
AIM EXPRESS short listing entities	3139
application assertion	1226, 1236–1237
application object	521
ARM diagrams	4688
ARM EXPRESS listing entities	4502
mapping table	1818
functional_specification_definition	
AIM EXPRESS short listing entities	3140
application assertion	1237
application object	522
ARM EXPRESS listing entities	4502
mapping table	1821
functional_specification_to_requirement_occurrence_assignment	
ARM diagrams	4647
functional_terminal_group	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3915
AIM EXPRESS short listing entities	3141
application assertion	1237
application object	522
ARM diagrams	4840
ARM EXPRESS listing entities	4502
mapping table	1827

functional_terminal_group_assignment	
application assertion	1237
application object	523
ARM diagrams	4840
ARM EXPRESS listing entities	4502
mapping table	1827
functional_unit	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3916
AIM EXPRESS short listing entities	3142
application assertion	1204, 1236–1239, 1315, 1325
application object	524
ARM diagrams	4680
ARM EXPRESS listing entities	4502
mapping table	1759, 1762–1763, 1769, 1828, 2763
functional_unit_definition	
application assertion	1230, 1237
application object	527
ARM diagrams	4681
ARM EXPRESS listing entities	4503
mapping table	1762
functional_unit_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4148
functional_unit_network_definition	
application assertion	1151, 1237–1238, 1325
application object	528
ARM diagrams	4681
ARM EXPRESS listing entities	4503
mapping table	1763
functional_unit_network_node_definition	
application assertion	1151, 1169, 1238–1239
application object	529
ARM diagrams	4680
ARM EXPRESS listing entities	4503
mapping table	1764
functional_unit_network_node_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4149
functional_unit_network_terminal_definition_bus_assignment	
application assertion	1238
application object	531
ARM diagrams	4681
ARM EXPRESS listing entities	4503
mapping table	1765
functional_unit_network_terminal_definition_node_assignment	
application assertion	1238
application object	532

ARM diagrams	4682
ARM EXPRESS listing entities	4503
mapping table	1766
functional_unit_requirement_allocation	
application assertion	1238–1239, 1301
application object	533
ARM diagrams	4648
ARM EXPRESS listing entities	4504
mapping table	1816
functional_unit_terminal	
application assertion	1239, 1325, 1347
application object	534
ARM diagrams	4682
ARM EXPRESS listing entities	4504
mapping table	1767
functional_unit_terminal_bus_assignment	
application assertion	1239
application object	535
ARM diagrams	4682
ARM EXPRESS listing entities	4504
mapping table	1768
functional_unit_terminal_bus_assignment_constraint	
AIM EXPRESS expanded listing rules	4150
functional_unit_terminal_definition	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3916
AIM EXPRESS short listing entities	3144
mapping table	1770, 1828, 1830
functional_unit_terminal_node_assignment	
application assertion	1239, 1342
application object	536
ARM diagrams	4683
ARM EXPRESS listing entities	4504
mapping table	1768
functional_unit_unique_constraint	
AIM EXPRESS expanded listing rules	4151
functional_unit_usage_view	
application assertion	1162–1163, 1237, 1240, 1249, 1279, 1309
application object	537
ARM diagrams	4689
ARM EXPRESS listing entities	4504
mapping table	1769, 1828
functional_unit_usage_view_terminal_definition	
application assertion	1154, 1197, 1230, 1238–1240
application object	537
ARM diagrams	4683

ARM EXPRESS listing entities	4504
mapping table	1770, 1828
functional_unit_usage_view_terminal_definition_make_from_relationship	
application assertion	1240
application object	539
ARM EXPRESS listing entities	4505
mapping table	1829
functional_unit_usage_view_terminal_definition_unique_constraint	
AIM EXPRESS expanded listing rules	4152
functional_usage_view_to_part_terminal_assignment	
application assertion	1240, 1342
application object	540
ARM diagrams	4685
ARM EXPRESS listing entities	4505
mapping table	1778, 1792, 1797
functional_usage_view_to_part_terminal_assignment_constraint	
AIM EXPRESS expanded listing rules	4153
functional_usage_view_to_printed_part_template_terminal_assignment	
application assertion	1241
application object	541
ARM diagrams	4685
ARM EXPRESS listing entities	4505
mapping table	1802
functionality	10
functionally_defined_transformation	
AIM diagrams	4903
AIM EXPRESS expanded listing entities	3917
gate_path_swap_specification	
application assertion	1241
application object	542
ARM diagrams	4676
ARM EXPRESS listing entities	4505
mapping table	1746, 1754
gbsf_check_curve	
AIM EXPRESS expanded listing functions	4273
application interpreted constructs	4332
gbsf_check_point	
AIM EXPRESS expanded listing functions	4273
application interpreted constructs	4329
gbsf_check_surface	
AIM EXPRESS expanded listing functions	4275
application interpreted constructs	4337
geometric_alignment	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3917
application object	543

ARM diagrams	4692
ARM EXPRESS listing entities	4505
mapping table	1873
geometric_curve_set	
AIM diagrams	4918
AIM EXPRESS expanded listing entities	3917
geometric_intersection	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3917
application object	544
ARM diagrams	4692
ARM EXPRESS listing entities	4505
mapping table	1873
geometric_or_dimension_select	
ARM diagrams	4725
ARM EXPRESS listing types	4409
geometric_representation_context	
AIM diagrams	4902
AIM EXPRESS expanded listing entities	3917
mapping table	2835–2836
geometric_representation_item	
AIM diagrams	4902
AIM EXPRESS expanded listing entities	3918
mapping table	1658, 2857
geometric_representation_relationship	
application assertion	1241
application object	544
ARM EXPRESS listing entities	4506
mapping table	2847
geometric_representation_relationship_purpose	
ARM EXPRESS listing types	4409
geometric_set	
AIM diagrams	4918
AIM EXPRESS expanded listing entities	3918
geometric_set_select	
AIM diagrams	4918
AIM EXPRESS expanded listing types	3818
geometric_status	
ARM EXPRESS listing types	4409
geometric_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3918
AIM EXPRESS short listing entities	3490
application assertion	1241–1243, 1325, 1329, 1344
application object	545
ARM diagrams	4720

ARM EXPRESS listing entities	4506
mapping table	1873
geometric_tolerance_group	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	3919
AIM EXPRESS short listing entities	3145
mapping table	1916, 1921
geometric_tolerance_qualifier	
application assertion	1243
application object	548
ARM diagrams	4721
ARM EXPRESS listing entities	4506
mapping table	1878
geometric_tolerance_qualifier_constraint	
AIM EXPRESS expanded listing rules	4156
geometric_tolerance_qualifiers	
ARM diagrams	4721
ARM EXPRESS listing types	4409
geometric_tolerance_relationship	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3920
geometric_tolerance_with_applied_material_condition_property	
application object	548
ARM diagrams	4720
ARM EXPRESS listing entities	4506
mapping table	1878
geometric_tolerance_with_referenced_datum_system	
application assertion	1243, 1322
application object	550
ARM diagrams	4691
ARM EXPRESS listing entities	4507
mapping table	1878
geometric_tolerance_with_specified_datum_system	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3920
AIM EXPRESS short listing entities	3148
mapping table	1838, 1840, 1842, 1878, 1903, 1910–1911
geometric_tolerance_without_referenced_datum_system	
application object	551
ARM diagrams	4696
ARM EXPRESS listing entities	4507
mapping table	1879
geometrically_bounded_2d_wireframe	
application object	552
ARM diagrams	4827
ARM EXPRESS listing entities	4507

mapping table	1941
geometrically_bounded_2d_wireframe_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3922
mapping table	1941
geometrically_bounded_surface	
application object	552
ARM diagrams	4823
ARM EXPRESS listing entities	4507
mapping table	2449
geometrically_bounded_surface_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3923
application interpreted constructs	4326
mapping table	2449
geometrically_bounded_wireframe_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3924
get_basis_surface	
AIM EXPRESS expanded listing functions	4276
get_cit	
ARM EXPRESS listing functions	4611
get_pctd	
ARM EXPRESS listing functions	4612
get_pctr	
ARM EXPRESS listing functions	4612
global_uncertainty_assigned_context	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3925
mapping table	2835
global_unit_assigned_context	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3925
AIM EXPRESS short listing entities	3490
mapping table	2835
group	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3925
AIM EXPRESS short listing entities	3490
mapping table	1713, 1827, 2746, 2800
group_assigned_item	
AIM diagrams	4885
AIM EXPRESS expanded listing types	3818
AIM EXPRESS short listing types	2956
group_assignment	
AIM diagrams	4885

AIM EXPRESS expanded listing entities	3925
group_product_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3925
AIM EXPRESS short listing entities	3150
mapping table	1467, 2313
group_relationship	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3925
group_shape_aspect	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3927
AIM EXPRESS short listing entities	3151
mapping table	1477, 2265, 2313
grouped_requirements_property	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3928
AIM EXPRESS short listing entities	3155
mapping table	1450, 1453, 2276, 2284, 2298
guided_wave_bare_die_terminal	
application object	552
ARM diagrams	4658
ARM EXPRESS listing entities	4507
mapping table	1631
guided_wave_terminal	
AIM EXPRESS short listing entities	3161
application object	553
ARM diagrams	4790
ARM EXPRESS listing entities	4507
mapping table	2457
half_space_solid	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3928
hiding_or_blanking_select	
AIM diagrams	4929
AIM EXPRESS expanded listing types	3819
hierarchical_decomposition	10
hierarchy	10
horizontal_justification	
ARM diagrams	4752
ARM EXPRESS listing types	4409
hour_in_day	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3819
hyperbola	
AIM diagrams	4914

AIM EXPRESS expanded listing entities	3929
identifier	
AIM diagrams	4886
AIM EXPRESS expanded listing types	3819
illuminance_data_element	
application object	553
ARM EXPRESS listing entities	4507
mapping table	1646
illuminance_measure_with_unit	
AIM EXPRESS short listing entities	3161
mapping table	1646
inductance_data_element	
application object	554
ARM EXPRESS listing entities	4507
mapping table	1646
inductance_measure_with_unit	
AIM EXPRESS short listing entities	3162
mapping table	1646
inspection_requirement_occurrence	
application object	555
ARM EXPRESS listing entities	4508
mapping table	2812
insulating substrate materials	10
integral_shield	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3929
AIM EXPRESS short listing entities	3162
mapping table	2185
integrated circuit	10
inter_stratum_extent	
application assertion	1184, 1201, 1244, 1252, 1258, 1292, 1319, 1355
application object	555
ARM diagrams	4750
ARM EXPRESS listing entities	4508
mapping table	2110
inter_stratum_extent_constraint	
AIM EXPRESS expanded listing rules	4156
inter_stratum_feature	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3932
AIM EXPRESS short listing entities	3163
application assertion	1177, 1203, 1206, 1246, 1252–1253, 1264, 1317, 1327, 1331
application object	556
ARM diagrams	4705
ARM EXPRESS listing entities	4508
mapping table	1879, 2094, 2098, 2110, 2116, 2162, 2164, 2208, 2212, 2214

inter_stratum_feature_edge_segment_template	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3932
AIM EXPRESS short listing entities	3168
application assertion	1195, 1247, 1252
application object	557
ARM diagrams	4834
ARM EXPRESS listing entities	4508
mapping table	2113
inter_stratum_feature_edge_template	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3932
AIM EXPRESS short listing entities	3170
application assertion	1252
application object	559
ARM diagrams	4636
ARM EXPRESS listing entities	4509
mapping table	2115
inter_stratum_feature_template	
application assertion	1252–1253
application object	560
ARM diagrams	4731
ARM EXPRESS listing entities	4509
mapping table	1947, 2424
inter_stratum_join_implementation	
application assertion	1253
application object	560
ARM diagrams	4737
ARM EXPRESS listing entities	4509
mapping table	2024
inter_stratum_join_relationship	
application assertion	1253, 1255
application object	562
ARM diagrams	4737
ARM EXPRESS listing entities	4509
mapping table	2025
interconnect_component_interface_terminal	
application assertion	1158, 1211, 1213, 1243
application object	562
ARM diagrams	4643
ARM EXPRESS listing entities	4509
mapping table	1524
interconnect_component_interface_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4157
interconnect_component_join_terminal	
application assertion	1158, 1211, 1213, 1243

application object	564
ARM diagrams	4643
ARM EXPRESS listing entities	4509
mapping table	1525
interconnect_component_join_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4158
interconnect_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3933
AIM EXPRESS short listing entities	3170
mapping table	2163, 2234
interconnect_module	
application assertion	1231, 1243, 1245, 1248, 1255, 1292, 1295
application object	565
ARM diagrams	4805
ARM EXPRESS listing entities	4510
mapping table	1999, 2115, 2375, 2823
interconnect_module_cavity_surface	
application assertion	1243
application object	569
ARM EXPRESS listing entities	4510
mapping table	2217, 2542
interconnect_module_component	
application assertion	1244
application object	570
ARM diagrams	4625
ARM EXPRESS listing entities	4510
mapping table	1418, 2375, 2824
interconnect_module_component_stratum_based_terminal	
application assertion	1244, 1248
application object	571
ARM diagrams	4823
ARM EXPRESS listing entities	4510
mapping table	2825
interconnect_module_component_surface_feature	
AIM EXPRESS short listing entities	3171
application assertion	1159, 1244, 1272
application object	571
ARM diagrams	4638
ARM EXPRESS listing entities	4510
mapping table	1449, 1508, 1526, 2063, 2264, 2376, 2826
interconnect_module_component_terminal	
application assertion	1159, 1244
application object	572
ARM diagrams	4643
ARM EXPRESS listing entities	4510

mapping table	1527
interconnect_module_constraint_region	
application assertion	1244–1246
application object	574
ARM diagrams	4762
ARM EXPRESS listing entities	4511
mapping table	2265
interconnect_module_cutout_segment_surface	
AIM EXPRESS short listing entities	3173
application assertion	1246–1247
application object	576
ARM EXPRESS listing entities	4511
mapping table	2218, 2543
interconnect_module_cutout_surface	
application assertion	1247
application object	578
ARM EXPRESS listing entities	4511
mapping table	2219, 2544
interconnect_module_design_object_category	
AIM diagrams	4863
AIM EXPRESS expanded listing entities	3933
AIM EXPRESS short listing entities	3172
ARM diagrams	4762
ARM EXPRESS listing types	4410
mapping table	2265
interconnect_module_design_object_select	
ARM diagrams	4763
ARM EXPRESS listing types	4410
interconnect_module_edge	
application assertion	1247
application object	578
ARM diagrams	4705
ARM EXPRESS listing entities	4511
mapping table	2116, 2212
interconnect_module_edge_segment	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3934
AIM EXPRESS short listing entities	3174
application assertion	1247
application object	579
ARM diagrams	4760
ARM EXPRESS listing entities	4511
mapping table	2116, 2213
interconnect_module_edge_segment_surface	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3934

AIM EXPRESS short listing entities	3176
application assertion	1247
application object	580
ARM diagrams	4695
ARM EXPRESS listing entities	4512
mapping table	1735, 2219, 2544
interconnect_module_edge_surface	
application assertion	1247
application object	581
ARM diagrams	4673
ARM EXPRESS listing entities	4512
mapping table	2221, 2546
interconnect_module_interface_terminal	
AIM EXPRESS short listing entities	3177
application object	582
ARM EXPRESS listing entities	4512
mapping table	2221
interconnect_module_join_terminal	
AIM EXPRESS short listing entities	3177
application assertion	1248
application object	582
ARM EXPRESS listing entities	4512
mapping table	2221
interconnect_module_macro_component	
application assertion	1248
application object	583
ARM EXPRESS listing entities	4512
mapping table	1419, 1528, 2117
interconnect_module_macro_component_join_terminal	
AIM EXPRESS short listing entities	3178
application assertion	1203, 1248
application object	584
ARM EXPRESS listing entities	4512
mapping table	2118
interconnect_module_primary_surface	
application object	585
ARM diagrams	4673
ARM EXPRESS listing entities	4513
mapping table	1736, 2221, 2546
interconnect_module_product_assembly_view_fabrication_joint	
application assertion	1248
application object	585
ARM diagrams	4672
ARM EXPRESS listing entities	4513
mapping table	1728, 2827
interconnect_module_secondary_surface	

application object	586
ARM diagrams	4673
ARM EXPRESS listing entities	4513
mapping table	1736, 2221, 2546
interconnect_module_stratum_assembly_relationship	
application assertion	1248, 1336
application object	587
ARM diagrams	4733
ARM EXPRESS listing entities	4513
mapping table	1957, 2120, 2913
interconnect_module_stratum_assembly_relationship_constraint	
AIM EXPRESS expanded listing rules	4158
interconnect_module_stratum_based_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3934
AIM EXPRESS short listing entities	3179
application assertion	1244
application object	588
ARM diagrams	4761
ARM EXPRESS listing entities	4513
mapping table	2121, 2221, 2828
interconnect_module_surface_feature	
application assertion	1243–1244, 1248–1249
application object	590
ARM diagrams	4673
ARM EXPRESS listing entities	4514
mapping table	1736, 2222, 2546
interconnect_module_surface_feature_constraint	
AIM EXPRESS expanded listing rules	4160
interconnect_module_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3935
AIM EXPRESS short listing entities	3179
application assertion	1203, 1244, 1249, 1352
application object	591
ARM diagrams	4761
ARM EXPRESS listing entities	4514
mapping table	1887, 2223, 2548, 2571
interconnect_module_terminal_or_inter_stratum_feature	
ARM diagrams	4735
interconnect_module_terminal_surface_constituent_relationship	
application assertion	1249
application object	593
ARM EXPRESS listing entities	4514
mapping table	2224, 2547
interconnect_module_usage_view	

application assertion	1244, 1248–1250
application object	594
ARM diagrams	4791
ARM EXPRESS listing entities	4514
mapping table	2225
interconnect_or_assembly_module_select	
ARM diagrams	4734
ARM EXPRESS listing types	4411
interface	10
interface_access_component_shape_aspect	
AIM EXPRESS short listing entities	3181
mapping table	2122
interface_access_material_removal_component_shape_aspect	
AIM EXPRESS short listing entities	3181
mapping table	2121
interface_access_material_removal_laminate_component	
application assertion	1250
application object	596
ARM EXPRESS listing entities	4515
mapping table	2121
interface_access_stratum_feature_template_component	
application assertion	1250
application object	597
ARM EXPRESS listing entities	4515
mapping table	2122
interface_component	
application assertion	1179, 1250–1251
application object	598
ARM diagrams	4625
ARM EXPRESS listing entities	4515
mapping table	1420, 1484
interface_component_allocation	
application assertion	1251
application object	599
ARM diagrams	4648
ARM EXPRESS listing entities	4515
mapping table	1562
interface_component_allocation_constraint	
AIM EXPRESS expanded listing rules	4160
interface_component_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3935
AIM EXPRESS short listing entities	3181
mapping table	1420, 1484
interface_mounted_join	
AIM diagrams	4890

AIM EXPRESS expanded listing entities	3935
AIM EXPRESS short listing entities	3182
application assertion	1251, 1296
application object	600
ARM diagrams	4644
ARM EXPRESS listing entities	4515
mapping table	1529
interface_requirement_occurrence	
application assertion	1251
application object	601
ARM diagrams	4821
ARM EXPRESS listing entities	4515
mapping table	1554
interface_specification	
application object	602
ARM diagrams	4667
ARM EXPRESS listing entities	4515
mapping table	1715
interfaced_group_component	
application assertion	1250
application object	602
ARM diagrams	4627
ARM EXPRESS listing entities	4515
mapping table	1431, 1967
interfaced_group_component_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3936
AIM EXPRESS short listing entities	3183
mapping table	1431, 1967
interfacial_connection	
application object	603
ARM diagrams	4739
ARM EXPRESS listing entities	4516
mapping table	2123
internal_probe_access_area	
application assertion	1251–1252
application object	604
ARM diagrams	4750
ARM EXPRESS listing entities	4516
mapping table	2123, 2928
internal_stratum_access	
application assertion	1251–1252
application object	605
ARM diagrams	4751
ARM EXPRESS listing entities	4516
mapping table	2124

internal_stratum_access_unique_constraint	
AIM EXPRESS expanded listing rules	4161
intersection_curve	
AIM diagrams	4912
AIM EXPRESS expanded listing entities	3936
intra_stratum_join_relationship	
application assertion	1188, 1190, 1253
application object	606
ARM diagrams	4737
ARM EXPRESS listing entities	4516
mapping table	2026
invisibility_context	
AIM diagrams	4866
AIM EXPRESS expanded listing types	3819
invisible_item	
AIM diagrams	4931
AIM EXPRESS expanded listing types	3819
item_defined_transformation	
AIM diagrams	4903
AIM EXPRESS expanded listing entities	3936
mapping table	2339
item_in_context	
AIM EXPRESS expanded listing functions	4277
item_restricted_requirement_occurrence	
application assertion	1180, 1253–1255
application object	607
ARM diagrams	4620
ARM EXPRESS listing entities	4516
mapping table	1450, 2276
join_relationship	
application assertion	1255, 1295
application object	608
ARM diagrams	4737
ARM EXPRESS listing entities	4516
mapping table	2026
join_shape_aspect	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3937
AIM EXPRESS short listing entities	3184
mapping table	2025–2026, 2045
join_two_physical_connectivity_definition_supporting_inter_stratum_feature	
application assertion	1255
application object	609
ARM EXPRESS listing entities	4517
mapping table	2125
join_two_physical_connectivity_definition_supporting_printed_component	

application assertion	1255
application object	610
ARM diagrams	4751
ARM EXPRESS listing entities	4517
mapping table	2127
junction_vertex_allocation	
application assertion	1255
application object	610
ARM diagrams	4738
ARM EXPRESS listing entities	4517
mapping table	2027
junction_vertex_allocation_constraint	
AIM EXPRESS expanded listing rules	4161
junction_vertex_allocation_unique_constraint	
AIM EXPRESS expanded listing rules	4162
keepout_design_object_category	
AIM EXPRESS short listing entities	3186
keepout_product_design_object_category	
ARM EXPRESS listing types	4411
keepout_purpose	
ARM EXPRESS listing types	4411
knot_type	
AIM diagrams	4915
AIM EXPRESS expanded listing types	3819
label	
AIM diagrams	4929
AIM EXPRESS expanded listing types	3819
ARM diagrams	4717
ARM EXPRESS listing types	4411
laminate_component	
application assertion	1177, 1205, 1246, 1253, 1256, 1264, 1317, 1327, 1330
application object	612
ARM diagrams	4744
ARM EXPRESS listing entities	4517
mapping table	2127
laminate_component_feature	
application object	613
ARM EXPRESS listing entities	4517
mapping table	2130
laminate_component_interface_terminal	
AIM EXPRESS short listing entities	3187
application assertion	1250
application object	614
ARM EXPRESS listing entities	4518
mapping table	1508, 1521, 2063, 2075, 2078, 2130–2131, 2137, 2150, 2176
laminate_component_join_terminal	

application assertion	1295
application object	614
ARM EXPRESS listing entities	4518
mapping table	2131
laminate_group_component_make_from_relationship	
AIM EXPRESS short listing entities	3191
application assertion	1256
application object	615
ARM EXPRESS listing entities	4518
mapping table	1515, 2074, 2131
laminate_text_component	
application assertion	1256
application object	616
ARM diagrams	4743
ARM EXPRESS listing entities	4518
mapping table	2133
laminate_text_string_component	
application assertion	1175
application object	618
ARM diagrams	4752
ARM EXPRESS listing entities	4518
mapping table	2135
land	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	3941
AIM EXPRESS short listing entities	3192
application assertion	1245, 1256–1257, 1263
application object	620
ARM diagrams	4773
ARM EXPRESS listing entities	4519
mapping table	2382, 2384, 2386–2387, 2389–2391, 2393–2394, 2396, 2398, 2400–2401
land_interface_terminal	
application assertion	1257
application object	622
ARM diagrams	4752
ARM EXPRESS listing entities	4519
mapping table	2137
land_join_terminal	
application assertion	1232, 1257, 1261
application object	623
ARM diagrams	4752
ARM EXPRESS listing entities	4519
mapping table	2139
land_physical_template	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3944

AIM EXPRESS short listing entities	3199
application assertion	1256–1257, 1274
application object	624
ARM diagrams	4784
ARM EXPRESS listing entities	4519
mapping table	2407–2408, 2412–2413, 2416–2417, 2424
land_physical_template_unique_constraint	
AIM EXPRESS expanded listing rules	4163
land_template_interface_terminal	
application assertion	1257, 1275
application object	627
ARM diagrams	4785
ARM EXPRESS listing entities	4519
mapping table	2425
land_template_join_terminal	
application assertion	1257, 1275
application object	629
ARM diagrams	4785
ARM EXPRESS listing entities	4520
mapping table	2426
land_template_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3944
AIM EXPRESS short listing entities	3205
application assertion	1175, 1257, 1350, 1352
application object	629
ARM diagrams	4785
ARM EXPRESS listing entities	4520
mapping table	2425–2426
land_template_terminal_class	
ARM diagrams	4785
ARM EXPRESS listing types	4412
land_template_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4164
language_reference_manual	
application assertion	1153
application object	632
ARM diagrams	4633
ARM EXPRESS listing entities	4520
mapping table	1394, 1674, 1715
layer	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3945
AIM EXPRESS short listing entities	3206
application assertion	1257–1258, 1272–1273, 1306
application object	632

ARM diagrams	4753
ARM EXPRESS listing entities	4520
mapping table	2141, 2879
layer_connection_point	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3945
AIM EXPRESS short listing entities	3207
application assertion	1187, 1190, 1253, 1258
application object	634
ARM diagrams	4735
ARM EXPRESS listing entities	4521
mapping table	2016, 2019, 2028
layer_connection_point_link	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	3946
AIM EXPRESS short listing entities	3211
application assertion	1186, 1258, 1349
application object	636
ARM diagrams	4722
ARM EXPRESS listing entities	4521
mapping table	2030
layer_position	
ARM diagrams	4784
layer_position_type	
ARM EXPRESS listing types	4412
layer_purpose	
ARM diagrams	4784
ARM EXPRESS listing types	4412
layer_qualified_layout_spacing_requirement_occurrence	
application assertion	1258
application object	638
ARM diagrams	4632
ARM EXPRESS listing entities	4521
mapping table	2284
layered_item	
AIM diagrams	4867
AIM EXPRESS expanded listing types	3819
layout_land_width_tolerance_requirement_occurrence	
application assertion	1258–1259
application object	639
ARM EXPRESS listing entities	4521
mapping table	2239, 2813
layout_line_width_tolerance_requirement_occurrence	
application assertion	1259
application object	640
ARM EXPRESS listing entities	4521

mapping table	2240, 2814
layout_macro_component	
AIM EXPRESS short listing entities	3212
application assertion	1259
application object	641
ARM EXPRESS listing entities	4522
mapping table	2142
layout_macro_definition	
AIM EXPRESS short listing entities	3214
application assertion	1259
application object	642
ARM EXPRESS listing entities	4522
mapping table	2143
layout_macro_definition_terminal_to_usage_terminal_assignment	
AIM EXPRESS short listing entities	3215
application assertion	1260–1261
application object	643
ARM EXPRESS listing entities	4522
mapping table	2144, 2593
layout_macro_floor_plan_template	
AIM EXPRESS short listing entities	3216
application assertion	1259, 1261
application object	644
ARM EXPRESS listing entities	4522
mapping table	2147, 2595
layout_spacing_requirement_non_conforming_design_object	
application assertion	1261–1264
application object	646
ARM diagrams	4628
ARM EXPRESS listing entities	4522
mapping table	2285
layout_spacing_requirement_occurrence	
application assertion	1229, 1261, 1265, 1343
application object	647
ARM diagrams	4632
ARM EXPRESS listing entities	4523
mapping table	1453, 2298
layout_spacing_requirement_occurrence_unique_constraint	
AIM EXPRESS expanded listing rules	4167
lead_form_specification	
application assertion	1323
application object	650
ARM diagrams	4670
ARM EXPRESS listing entities	4523
mapping table	1715
lead_form_type	

ARM EXPRESS listing types	4412
leap_year	
AIM EXPRESS expanded listing functions	4277
length_data_element	
application assertion	1164, 1166, 1168–1169, 1172, 1179–1180, 1189, 1194, 1196, 1201–1202, 1210, 1229, 1250, 1258–1259, 1265–1266, 1272–1273, 1277, 1281, 1286–1287, 1291–1292, 1298, 1303–1304, 1308, 1310, 1313–1314, 1329, 1338, 1345, 1347, 1359
application object	650
ARM diagrams	4659
ARM EXPRESS listing entities	4523
mapping table	1646
length_measure	
AIM diagrams	4895
AIM EXPRESS expanded listing types	3819
length_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3946
mapping table	1646
length_trimmed_terminal	
AIM EXPRESS short listing entities	3217
application assertion	1265
application object	651
ARM diagrams	4789
ARM EXPRESS listing entities	4523
mapping table	2457
length_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3946
library_defined_assembly_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3947
AIM EXPRESS short listing entities	3219
mapping table	1550
library_defined_bare_die	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3947
AIM EXPRESS short listing entities	3220
mapping table	1618
library_defined_functional_unit	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3948
AIM EXPRESS short listing entities	3221
mapping table	1762, 1769, 1828, 2404, 2763
library_defined_interconnect_definition	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3948

AIM EXPRESS short listing entities	3222
mapping table	2234
library_defined_model	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	3948
library_defined_package	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3949
AIM EXPRESS short listing entities	3223
mapping table	1366, 1372, 2450, 2458
library_defined_packaged_connector	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3949
AIM EXPRESS short listing entities	3224
mapping table	1548, 2498
library_defined_packaged_part	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3950
AIM EXPRESS short listing entities	3225
mapping table	1377, 2496, 2500, 2524
library_defined_physical_unit	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3950
AIM EXPRESS short listing entities	3227
mapping table	1542, 2225, 2404, 2670, 2677, 2763
library_defined_product_definition	
AIM EXPRESS short listing entities	3228
library_ee_product_definition	
application object	652
ARM diagrams	4777
ARM EXPRESS listing entities	4523
mapping table	2404
library_item	
ARM diagrams	4777
library_model	
ARM diagrams	4616
life_cycle_stage	
ARM diagrams	4815
ARM EXPRESS listing types	4412
limit_condition	
AIM diagrams	4897
AIM EXPRESS expanded listing types	3820
limit_dimension_representation	
application assertion	1265
application object	652
ARM diagrams	4719

ARM EXPRESS listing entities	4523
mapping table	1882
line	
AIM diagrams	4906
AIM EXPRESS expanded listing entities	3951
line_width_class	
application assertion	1259, 1266
application object	653
ARM diagrams	4785
ARM EXPRESS listing entities	4523
mapping table	2428
linear_array_component_definition_link	
AIM EXPRESS short listing entities	3228
mapping table	1437, 1976
linear_array_component_shape_aspect_link	
AIM EXPRESS short listing entities	3229
mapping table	1976
linear_array_placement_group_component	
application assertion	1265–1266
application object	654
ARM EXPRESS listing entities	4524
mapping table	1431, 1968
linear_array_placement_group_component_definition	
AIM EXPRESS short listing entities	3231
mapping table	1431, 1968
linear_array_placement_group_component_link	
application assertion	1266, 1314
application object	655
ARM EXPRESS listing entities	4524
mapping table	1437, 1976
linear_array_placement_group_component_shape_aspect	
AIM EXPRESS short listing entities	3231
mapping table	1968
linear_composite_array_shape_aspect	
AIM EXPRESS short listing entities	3231
mapping table	2561
linear_composite_array_shape_aspect_link	
AIM EXPRESS short listing entities	3232
mapping table	2565
linear_extension	
application object	656
ARM diagrams	4692
ARM EXPRESS listing entities	4524
mapping table	1883
linear_profile_tolerance	
AIM diagrams	4897

AIM EXPRESS expanded listing entities	3952
AIM EXPRESS short listing entities	3233
mapping table	1840–1841, 1910–1911
list_characteristic	
application assertion	1172, 1266
ARM EXPRESS listing types	4412
list_face_loops	
AIM EXPRESS expanded listing functions	4277
list_of_reversible_topology_item	
AIM diagrams	4922
AIM EXPRESS expanded listing types	3820
list_of_topology_reversed	
AIM EXPRESS expanded listing functions	4278
list_to_array	
AIM EXPRESS expanded listing functions	4278
list_to_set	
AIM EXPRESS expanded listing functions	4278
local_time	
AIM diagrams	4883
AIM EXPRESS expanded listing entities	3952
mapping table	1612
located_interconnect_module_thickness_requirement	
application assertion	1249, 1266
application object	657
ARM diagrams	4762
ARM EXPRESS listing entities	4524
mapping table	2232
locating_or_refinement_linear_profile_tolerance	
ARM diagrams	4693
ARM EXPRESS listing types	4413
locating_or_refinement_surface_profile_tolerance	
ARM diagrams	4696
ARM EXPRESS listing types	4413
locating_or_relating_position_tolerance	
ARM diagrams	4697
ARM EXPRESS listing types	4413
logical_property	
ARM diagrams	4660
ARM EXPRESS listing types	4413
loop	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3952
lot_effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	3952
mapping table	2726

luminous_flux_data_element	
application object	658
ARM EXPRESS listing entities	4524
mapping table	1646
luminous_flux_measure_with_unit	
AIM EXPRESS short listing entities	3237
mapping table	1646
luminous_intensity_data_element	
application object	659
ARM EXPRESS listing entities	4524
mapping table	1646
luminous_intensity_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3820
luminous_intensity_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3952
mapping table	1646
luminous_intensity_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3953
magnetic_flux_data_element	
application object	659
ARM EXPRESS listing entities	4525
mapping table	1646
magnetic_flux_density_data_element	
application object	660
ARM EXPRESS listing entities	4525
mapping table	1646
magnetic_flux_density_measure_with_unit	
AIM EXPRESS short listing entities	3238
mapping table	1646
magnetic_flux_measure_with_unit	
AIM EXPRESS short listing entities	3239
mapping table	1646
make_array_of_array	
AIM EXPRESS expanded listing functions	4279
make_from_connectivity_relationship	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	3953
AIM EXPRESS short listing entities	3239
mapping table	1813, 2165
make_from_feature_relationship	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	3954
AIM EXPRESS short listing entities	3240

mapping table	1513, 2069
make_from_functional_unit_terminal_definition_relationship	
AIM EXPRESS short listing entities	3242
mapping table	1829
make_from_model_port_relationship	
AIM EXPRESS short listing entities	3243
mapping table	1391
make_from_part_feature_relationship	
AIM EXPRESS short listing entities	3245
mapping table	1381
make_from_usage_option	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	3954
AIM EXPRESS short listing entities	3491
mapping table	1515, 2074, 2204
managed_design_object	
AIM diagrams	4872
AIM EXPRESS expanded listing types	3821
AIM EXPRESS short listing types	2956
application assertion	1150, 1170, 1203
application object	661
ARM diagrams	4842
ARM EXPRESS listing entities	4530
mapping table	1697
mandatory_product_related_product_category_value	
AIM EXPRESS expanded listing rules	4171
manifold_constraining_context_dependent_shape_representation	
AIM EXPRESS short listing entities	3247
mapping table	2337
manifold_solid_brep	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	3954
manifold_subsurface_shape	
application object	678
ARM EXPRESS listing entities	4530
mapping table	2849
manifold_subsurface_shape_representation	
mapping table	2321, 2849
manifold_surface_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	3959
application interpreted constructs	4344
mapping table	2319, 2349, 2360–2361, 2363, 2370, 2644
mapped_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3959

AIM EXPRESS short listing entities	3491
mapping table ... 1575, 1582, 1585, 2193, 2195, 2198, 2201, 2322, 2330, 2334, 2347, 2349, 2368, 2515, 2518, 2574, 2619, 2632, 2641, 2679, 2750, 2847, 2854, 2898, 2900, 2906–2907, 2914, 2917	
markable_item	
ARM diagrams	4785
ARM EXPRESS listing types	4413
marked_message_select	
ARM diagrams	4786
ARM EXPRESS listing types	4413
marking	
application assertion	1266–1268
application object	679
ARM diagrams	4786
ARM EXPRESS listing entities	4531
mapping table	1488, 1704, 2010
marking_constraint	
AIM EXPRESS expanded listing rules	4171
mass_data_element	
application object	681
ARM EXPRESS listing entities	4531
mapping table	1647
mass_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3821
mass_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3960
mapping table	1647
mass_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3960
material_addition_feature	
application assertion	1268
application object	682
ARM diagrams	4836
ARM EXPRESS listing entities	4531
mapping table	2903
material_addition_feature_template	
application assertion	1268, 1283, 1341
application object	682
ARM diagrams	4755
ARM EXPRESS listing entities	4531
mapping table	2596
material_assembly_relationship	
application assertion	1268, 1281
ARM diagrams	4788

material_assembly_relationship_constraint	
AIM EXPRESS expanded listing rules	4172
material_assembly_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4173
material_composition_relationship	
application assertion	1268–1269
application object	683
ARM diagrams	4734
ARM EXPRESS listing entities	4531
mapping table	1958, 2447
material_condition	
ARM diagrams	4801
ARM EXPRESS listing types	4414
material_condition_property_names	
ARM diagrams	4720
ARM EXPRESS listing types	4414
material_conductivity_classification	
ARM diagrams	4821
ARM EXPRESS listing types	4414
material_definition	
application object	684
ARM EXPRESS listing entities	4531
mapping table	2448
material_designation	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	3960
AIM EXPRESS short listing entities	3492
mapping table	2443
material_designation_characterization	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	3960
material_electrical_conductivity_category	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3960
AIM EXPRESS short listing entities	3247
mapping table	2443, 2815
material_electrical_conductivity_requirement_occurrence	
application object	685
ARM diagrams	4821
ARM EXPRESS listing entities	4532
mapping table	2815
material_functional_category	
AIM EXPRESS short listing entities	3248
material_optical_conductivity_category	
AIM EXPRESS short listing entities	3248
mapping table	2443

material_or_specification	
ARM diagrams	4749
ARM EXPRESS listing types	4414
material_property	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	3961
mapping table	2677
material_property_representation	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	3961
material_removal_component_shape_aspect	
AIM EXPRESS short listing entities	3249
mapping table	2147
material_removal_feature	
application assertion	1269
application object	685
ARM diagrams	4836
ARM EXPRESS listing entities	4532
mapping table	2904
material_removal_feature_template	
application assertion	1269, 1341
application object	686
ARM diagrams	4755
ARM EXPRESS listing entities	4532
mapping table	2596
material_removal_laminate_component	
application assertion	1209, 1269
application object	689
ARM EXPRESS listing entities	4532
mapping table	2147
material_shape_modification	
application object	690
ARM diagrams	4836
ARM EXPRESS listing entities	4532
mapping table	2904
material_specification	
application object	690
ARM diagrams	4670
ARM EXPRESS listing entities	4532
mapping table	1715
material_state_change_enumeration	
ARM EXPRESS listing types	4414
material_thermal_conductivity_category	
AIM EXPRESS short listing entities	3249
mapping table	2444
mating_connector_component	

application assertion	1251, 1269–1270
application object	691
ARM diagrams	4631
ARM EXPRESS listing entities	4532
mapping table	1555
mating_connector_part	
ARM diagrams	4631
ARM EXPRESS listing types	4414
mating_connector_placement_relationship	
application assertion	1270
application object	692
ARM diagrams	4634
ARM EXPRESS listing entities	4533
mapping table	1556
mating_connector_placement_relationship_constraint	
AIM EXPRESS expanded listing rules	4173
mating_connector_termination	
application assertion	1270, 1342
application object	693
ARM diagrams	4634
ARM EXPRESS listing entities	4533
mapping table	1476
mating_connector_termination_constraint	
AIM EXPRESS expanded listing rules	4173
mating_connector_termination_unique_constraint	
AIM EXPRESS expanded listing rules	4174
max_data_element	
ARM EXPRESS listing functions	4612
maximum_dimension_representation	
application object	694
ARM diagrams	4718
ARM EXPRESS listing entities	4533
mapping table	1883
maximum_tolerance_characteristic	
application assertion	1271
application object	695
ARM diagrams	4661
ARM EXPRESS listing entities	4533
mapping table	1647
measure_orientation	
ARM diagrams	4672
ARM EXPRESS listing types	4415
measure_qualification	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	3961
AIM EXPRESS short listing entities	3493

application object	695
ARM EXPRESS listing entities	4533
mapping table	1647, 1863, 1878, 1907–1908, 1937–1938
measure_representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3961
mapping table	1392, 1394, 1648, 1650, 1669, 1674
measure_value	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3822
measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	3961
mapping table	1395, 1644, 1648, 1675
mechanical_requirement_occurrence	
application object	696
ARM EXPRESS listing entities	4533
mapping table	2816
message_content	
ARM diagrams	4752
ARM EXPRESS listing types	4415
message_source_select	
ARM diagrams	4786
ARM EXPRESS listing types	4415
message_type	
ARM diagrams	4786
ARM EXPRESS listing types	4415
metal_or_dielectric	
ARM EXPRESS listing types	4415
minimally_defined_bare_die_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3962
AIM EXPRESS short listing entities	3250
application assertion	1271, 1284
application object	696
ARM diagrams	4658
ARM EXPRESS listing entities	4534
mapping table	1631, 1887, 2548, 2571
minimally_defined_component_terminal	
application assertion	1271
application object	697
ARM diagrams	4642
ARM EXPRESS listing entities	4534
mapping table	1778, 1807
minimum_dimension_representation	
application object	697

ARM diagrams	4718
ARM EXPRESS listing entities	4534
mapping table	1883
minimum_tolerance_characteristic	
application assertion	1265, 1271
application object	698
ARM diagrams	4661
ARM EXPRESS listing entities	4534
mapping table	1647
minute_in_hour	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3822
mixed_loop_type_set	
AIM EXPRESS expanded listing functions	4280
model_parameter	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	3962
AIM EXPRESS short listing entities	3251
application assertion	1153, 1221, 1227, 1271, 1283
application object	698
ARM diagrams	4618
ARM EXPRESS listing entities	4534
mapping table	1394, 1648, 1674
model_parameter_type	
ARM diagrams	4618
ARM EXPRESS listing types	4415
model_parameter_unique_constraint	
AIM EXPRESS expanded listing rules	4175
model_parametermodel_parameter	
mapping table	1394, 1648, 1674
modified_geometric_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3963
AIM EXPRESS short listing entities	3493
mapping table	1835, 1878
month_in_year_number	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3822
month_representation	
ARM diagrams	4656
ARM EXPRESS listing types	4416
mounting_restriction_area	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3963
AIM EXPRESS short listing entities	3252
application assertion	1181, 1272, 1300

application object	700
ARM diagrams	4630
ARM EXPRESS listing entities	4534
mapping table	1465, 2310
mounting_restriction_volume	
AIM EXPRESS short listing entities	3253
application assertion	1180, 1272, 1300
application object	701
ARM EXPRESS listing entities	4534
mapping table	1466, 2311
mounting_technology_type	
ARM EXPRESS listing types	4416
movable_packaged_component_join_terminal	
application assertion	1213, 1272
application object	724
ARM EXPRESS listing entities	4538
mapping table	1530
msb_shells	
AIM EXPRESS expanded listing functions	4280
msf_curve_check	
AIM EXPRESS expanded listing functions	4282
application interpreted constructs	4351
msf_surface_check	
AIM EXPRESS expanded listing functions	4283
application interpreted constructs	4355
multi-layer printed circuit board	10
multi_layer_component_shape_aspect	
AIM EXPRESS short listing entities	3255
mapping table	2149
multi_layer_material_removal_component_shape_aspect	
AIM EXPRESS short listing entities	3255
mapping table	2148
multi_layer_material_removal_laminate_component	
application assertion	1272
application object	703
ARM EXPRESS listing entities	4535
mapping table	2148
multi_layer_stratum_feature	
AIM EXPRESS short listing entities	3256
application assertion	1272
application object	704
ARM EXPRESS listing entities	4535
mapping table	2148
multi_layer_stratum_feature_template_component	
application assertion	1273
application object	704

ARM EXPRESS listing entities	4535
mapping table	2149
named_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	3963
AIM EXPRESS short listing entities	3494
mapping table	1391, 1668
network_node_definition	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	3964
AIM EXPRESS short listing entities	3256
mapping table	1764
next_assembly_usage_occurrence	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	3964
AIM EXPRESS short listing entities	3494
next_assembly_usage_occurrencenext_assembly_usage_occurrence	
mapping table	1403–1404, 1501, 1532, 1955, 1959, 2149, 2377, 2822, 2828
next_higher_assembly_interface_requirement	
application assertion	1251, 1270, 1273
application object	705
ARM diagrams	4635
ARM EXPRESS listing entities	4535
mapping table	1558
next_higher_assembly_interface_requirement_constraint	
AIM EXPRESS expanded listing rules	4176
next_higher_assembly_relationship	
application assertion	1173–1174, 1180, 1273
application object	706
ARM diagrams	4621
ARM EXPRESS listing entities	4535
mapping table	1404, 1532, 1959, 2149, 2377, 2828
next_higher_assembly_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4177
no_shape_for_make_from	
AIM EXPRESS expanded listing rules	4177
nomenclature	10
ARM diagrams	4787
ARM EXPRESS listing types	4416
non_conductive_base_blind_via	
application object	708
ARM diagrams	4739
ARM EXPRESS listing entities	4535
mapping table	2150
non_conductive_cross_section_template	
application assertion	1273

application object	709
ARM EXPRESS listing entities	4536
mapping table	2728
non_feature_shape_definition	
application assertion	1273, 1351
application object	710
ARM EXPRESS listing entities	4536
mapping table	2849
non_feature_shape_element	
application assertion	1273–1274, 1350
application object	711
ARM diagrams	4710
ARM EXPRESS listing entities	4536
mapping table	1884
non_feature_shape_element_relationship	
application assertion	1274
application object	713
ARM EXPRESS listing entities	4537
mapping table	1886
non_functional_land	
application assertion	1246, 1264, 1274–1275
application object	714
ARM diagrams	4774
ARM EXPRESS listing entities	4537
mapping table	2393
non_functional_land_interface_terminal	
application assertion	1275
application object	716
ARM diagrams	4753
ARM EXPRESS listing entities	4537
mapping table	2150
non_functional_land_join_terminal	
application assertion	1232, 1261, 1275
application object	717
ARM diagrams	4753
ARM EXPRESS listing entities	4537
mapping table	2151
non_numeric_property_type	
ARM diagrams	4660
non_numeric_property_value	
ARM EXPRESS listing types	4416
normalise	
AIM EXPRESS expanded listing functions	4284
numeric_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3822

numeric_or_non_numeric_select	
ARM diagrams	4660
ARM EXPRESS listing types	4416
numeric_parameter	
application assertion	1271, 1275, 1302, 1314, 1340, 1344
application object	718
ARM diagrams	4660
ARM EXPRESS listing entities	4537
mapping table	1650
offset_curve_2d	
AIM diagrams	4913
AIM EXPRESS expanded listing entities	3964
offset_curve_3d	
AIM diagrams	4913
AIM EXPRESS expanded listing entities	3965
offset_surface	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	3965
offset_types	
ARM diagrams	4728
ARM EXPRESS listing types	4416
one_direction_repeat_factor	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	3965
open_shell	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3965
open_shell_based_surface	
application object	719
ARM diagrams	4827
ARM EXPRESS listing entities	4538
mapping table	2360
operational_requirement_occurrence_relationship	
application assertion	1275
application object	719
ARM diagrams	4822
ARM EXPRESS listing entities	4538
mapping table	2816
operational_requirement_occurrence_relationship_constraint	
AIM EXPRESS expanded listing rules	4177
opposing_boundary_dimensional_size	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	3965
AIM EXPRESS short listing entities	3257
mapping table	1869
opposing_boundary_set_size_characteristic	

application assertion	1275
application object	721
ARM diagrams	4694
ARM EXPRESS listing entities	4538
mapping table	1886
organization	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3965
application assertion	1188, 1193, 1219, 1221–1224, 1228, 1293, 1340, 1360–1361
application object	722
ARM diagrams	4761
ARM EXPRESS listing entities	4538
mapping table	1614, 2874
organization_assigned_item	
AIM diagrams	4878
AIM EXPRESS expanded listing types	3822
AIM EXPRESS short listing types	2958
organization_assignment	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3965
organization_relationship	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3966
organization_role	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	3966
AIM EXPRESS short listing entities	3494
organization_unique_constraint	
AIM EXPRESS expanded listing rules	4178
organizational_address	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3966
mapping table	1614, 2874
organizational_project	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3966
orientation	
application assertion	1163, 1169, 1196
ARM diagrams	4832
orientation_select	
ARM diagrams	4701
ARM EXPRESS listing types	4417
orientation_type	
ARM EXPRESS listing types	4417
oriented_closed_shell	
AIM diagrams	4923

AIM EXPRESS expanded listing entities	3966
oriented_edge	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	3967
oriented_face	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	3967
oriented_open_shell	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	3967
oriented_path	
AIM diagrams	4922
AIM EXPRESS expanded listing entities	3968
orthogonal_complement	
AIM EXPRESS expanded listing functions	4284
outer_boundary_curve	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	3968
owned_design_object	
application assertion	1276
application object	723
ARM EXPRESS listing entities	4538
mapping table	1700
owner_assigned_item	
AIM EXPRESS short listing types	2958
mapping table	1700
owner_assignment	
AIM EXPRESS short listing entities	3258
application assertion	1276
application object	723
ARM EXPRESS listing entities	4538
mapping table	1701
package	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3971
AIM EXPRESS short listing entities	3259
application assertion	1152, 1236, 1276–1277, 1279–1281
application object	725
ARM diagrams	4641
ARM EXPRESS listing entities	4539
mapping table	1366, 1372, 2450, 2458
package_body	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3971
AIM EXPRESS short listing entities	3265
application assertion	1280

application object	738
ARM diagrams	4792
ARM EXPRESS listing entities	4540
mapping table	1887, 2478, 2548
package_body_bottom_surface	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3971
AIM EXPRESS short listing entities	3266
application object	740
ARM diagrams	4722
ARM EXPRESS listing entities	4540
mapping table	2479
package_body_edge_segment_surface	
AIM EXPRESS short listing entities	3266
application assertion	1280
application object	740
ARM EXPRESS listing entities	4540
mapping table	2479
package_body_edge_surface	
AIM EXPRESS short listing entities	3268
application assertion	1280
application object	741
ARM EXPRESS listing entities	4540
mapping table	2480
package_body_surface	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3972
AIM EXPRESS short listing entities	3268
application assertion	1280, 1282, 1302, 1306, 1321, 1342
application object	741
ARM diagrams	4722
ARM EXPRESS listing entities	4540
mapping table	1887, 2480, 2548
package_body_top_surface	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	3972
AIM EXPRESS short listing entities	3269
application object	742
ARM diagrams	4722
ARM EXPRESS listing entities	4540
mapping table	2481
package_external_reference	
application assertion	1279, 1281
application object	742
ARM diagrams	4676
ARM EXPRESS listing entities	4541

mapping table	1779
package_external_reference_constraint	
AIM EXPRESS expanded listing rules	4178
package_or_package_external_reference	
ARM diagrams	4794
ARM EXPRESS listing types	4417
package_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3976
AIM EXPRESS short listing entities	3270
application assertion	1152, 1280–1282, 1302, 1353
application object	743
ARM diagrams	4790
ARM EXPRESS listing entities	4541
mapping table	1887, 2481, 2548
package_terminal_external_reference	
application assertion	1280–1281
application object	747
ARM diagrams	4677
ARM EXPRESS listing entities	4541
mapping table	1780
package_terminal_external_reference_constraint	
AIM EXPRESS expanded listing rules	4179
package_terminal_external_reference_unique_constraint	
AIM EXPRESS expanded listing rules	4179
package_terminal_occurrence	
application assertion	1158, 1282
application object	748
ARM diagrams	4644
ARM EXPRESS listing entities	4542
mapping table	1532
package_terminal_or_package_terminal_external_reference	
ARM diagrams	4795
ARM EXPRESS listing types	4417
package_terminal_surface_constituent_relationship	
application assertion	1282
application object	749
ARM EXPRESS listing entities	4542
mapping table	2487
package_unique_constraint	
AIM EXPRESS expanded listing rules	4180
packaged_component	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3977
AIM EXPRESS short listing entities	3273
application assertion	1212, 1229, 1277–1278, 1282

application object	750
ARM diagrams	4626
ARM EXPRESS listing entities	4542
mapping table	1421, 1423, 1484
packaged_component_join_terminal	
application assertion	1159, 1278
application object	751
ARM diagrams	4644
ARM EXPRESS listing entities	4542
mapping table	1533
packaged_connector	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	3978
AIM EXPRESS short listing entities	3276
application assertion	1162, 1269, 1278–1279
application object	752
ARM diagrams	4794
ARM EXPRESS listing entities	4542
mapping table	1548, 2498
packaged_connector_component	
application assertion	1278
application object	753
ARM diagrams	4626
ARM EXPRESS listing entities	4542
mapping table	1423, 1484
packaged_connector_component_interface_terminal	
application assertion	1158, 1251, 1278
application object	755
ARM diagrams	4645
ARM EXPRESS listing entities	4542
mapping table	1535
packaged_connector_terminal_relationship	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3979
AIM EXPRESS short listing entities	3278
application assertion	1279
application object	757
ARM diagrams	4793
ARM EXPRESS listing entities	4543
mapping table	2499, 2522
packaged_connector_terminal_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4182
packaged_or_printed_terminal	
ARM diagrams	4634
ARM EXPRESS listing types	4417
packaged_part	

AIM diagrams	4887
AIM EXPRESS expanded listing entities	3980
AIM EXPRESS short listing entities	3279
application assertion	1152, 1278–1279, 1284
application object	758
ARM diagrams	4794
ARM EXPRESS listing entities	4543
mapping table	1377, 2496, 2500, 2524
packaged_part_interface_terminal	
application assertion	1161, 1278–1279
application object	761
ARM diagrams	4795
ARM EXPRESS listing entities	4543
mapping table	2503, 2527
packaged_part_join_terminal	
application assertion	1278–1279
application object	762
ARM diagrams	4795
ARM EXPRESS listing entities	4543
mapping table	2504, 2527
packaged_part_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3981
AIM EXPRESS short listing entities	3281
application assertion	1214, 1229, 1270, 1279–1280
application object	763
ARM diagrams	4795
ARM EXPRESS listing entities	4544
mapping table	1887, 2503–2504, 2527, 2548, 2571
packaged_part_unique_constraint	
AIM EXPRESS expanded listing rules	4184
padstack_definition	
AIM EXPRESS short listing entities	3283
application assertion	1233, 1236, 1274, 1282–1283, 1335
application object	405, 764
ARM EXPRESS listing entities	4544
mapping table	2597, 2763
padstack_definition_shape	
application object	765
ARM EXPRESS listing entities	4544
mapping table	2630
padstack_location_in_footprint_definition	
application object	766
ARM EXPRESS listing entities	4544
mapping table	2604
padstack_location_in_footprint_definition_transform	

application object	767
ARM EXPRESS listing entities	4544
mapping table	2632
padstack_occurrence	
AIM EXPRESS short listing entities	3285
application object	767
ARM EXPRESS listing entities	4545
mapping table	2153
padstack_occurrence_shape_aspect_relationship	
AIM EXPRESS short listing entities	3285
mapping table	2154
padstack_occurrence_sub_assembly_relationship	
application object	768
ARM EXPRESS listing entities	4545
mapping table	2154
parabola	
AIM diagrams	4914
AIM EXPRESS expanded listing entities	3981
parallel_shape_element_relationship	
application object	769
ARM diagrams	4723
ARM EXPRESS listing entities	4545
mapping table	1887
parallelism_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3981
AIM EXPRESS short listing entities	3286
application object	770
ARM diagrams	4691
ARM EXPRESS listing entities	4545
mapping table	1887
parameter_assignment	
application assertion	1154, 1221, 1224, 1227, 1237, 1283
application object	771
ARM diagrams	4662
ARM EXPRESS listing entities	4545
mapping table	1651
parameter_assignment_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3981
AIM EXPRESS short listing entities	3287
mapping table	1651
parameter_type	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	3982
AIM EXPRESS short listing entities	3288

parameter_value	
AIM diagrams	4909
AIM EXPRESS expanded listing types	3822
parametric_representation_context	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	3982
AIM EXPRESS short listing entities	3495
part	11
part_array_feature	
application object	772
ARM diagrams	4798
ARM EXPRESS listing entities	4545
mapping table	2548
part_based_constraint_occurrence	
ARM diagrams	4821
part_connected_terminals_definition	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3982
AIM EXPRESS short listing entities	3289
application assertion	1284, 1295
application object	772
ARM diagrams	4754
ARM EXPRESS listing entities	4546
mapping table	2000, 2155, 2506, 2527
part_device	
application assertion	1214, 1284
application object	774
ARM diagrams	4797
ARM EXPRESS listing entities	4546
mapping table	2531
part_device_terminal	
application assertion	1214, 1284–1285
application object	775
ARM diagrams	4797
ARM EXPRESS listing entities	4546
mapping table	2532
part_device_terminal_model_port_assignment	
application assertion	1284–1285
application object	776
ARM diagrams	4793
ARM EXPRESS listing entities	4546
mapping table	2510, 2534
part_device_terminal_model_port_assignment_unique_constraint	
AIM EXPRESS expanded listing rules	4185
part_device_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4186

part_device_unique_constraint	
AIM EXPRESS expanded listing rules	4187
part_feature	
application assertion	1155, 1175, 1178, 1206, 1232, 1254, 1285–1286, 1316, 1327, 1330, 1350
application object	778
ARM diagrams	4706
ARM EXPRESS listing entities	4547
mapping table	1887, 2548
part_feature_constraint	
AIM EXPRESS expanded listing rules	4188
part_feature_make_from_relationship	
application assertion	1285
application object	780
ARM EXPRESS listing entities	4547
mapping table	1381
part_feature_model_port_assignment	
application assertion	1285, 1342
application object	782
ARM diagrams	4793
ARM EXPRESS listing entities	4547
mapping table	2510, 2535, 2586, 2688
part_feature_or_template_feature	
ARM diagrams	4637
ARM EXPRESS listing types	4417
part_feature_placement_link	
application assertion	1286
application object	783
ARM EXPRESS listing entities	4547
mapping table	2559
part_feature_unique_constraint	
AIM EXPRESS expanded listing rules	4190
part_group_feature	
application assertion	1232
application object	784
ARM diagrams	4798
ARM EXPRESS listing entities	4548
mapping table	2559
part_interface_access_feature	
AIM EXPRESS short listing entities	3290
application assertion	1286
application object	784
ARM EXPRESS listing entities	4548
mapping table	2559
part_linear_array_feature	
application assertion	1286
application object	786

ARM EXPRESS listing entities	4548
mapping table	2561
part_linear_array_feature_link	
application assertion	1286–1287
application object	786
ARM EXPRESS listing entities	4548
mapping table	2565
part_mating_feature	
AIM EXPRESS short listing entities	3290
application assertion	1286
application object	787
ARM EXPRESS listing entities	4549
mapping table	2566
part_mounting_feature	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3982
AIM EXPRESS short listing entities	3291
application assertion	1179, 1287, 1353
application object	788
ARM diagrams	4798
ARM EXPRESS listing entities	4549
mapping table	1887, 2548, 2566
part_rectangular_array_feature	
application assertion	1287
application object	789
ARM EXPRESS listing entities	4549
mapping table	2567
part_template	
application assertion	1156, 1256, 1287–1288, 1290–1291, 1315
application object	790
ARM diagrams	4755
ARM EXPRESS listing entities	4549
mapping table	2158, 2605
part_template_3d_keepout_shape	
application assertion	1288
application object	793
ARM EXPRESS listing entities	4550
mapping table	2635
part_template_3d_shape	
application assertion	1219, 1267, 1288–1289, 1351, 1356
application object	792
ARM diagrams	4801
ARM EXPRESS listing entities	4550
mapping table	2639
part_template_definition	
AIM diagrams	4888

AIM EXPRESS expanded listing entities	3986
AIM EXPRESS short listing entities	3292
mapping table ... 1730, 1732, 1899, 1942, 1945–1948, 2158, 2405–2406, 2424, 2441–2442, 2590, 2596, 2605, 2615, 2619, 2728–2730, 2738	
part_template_location_in_padstack_definition	
application object	795
ARM EXPRESS listing entities	4550
mapping table	2610
part_template_location_in_padstack_definition_transform	
application object	796
ARM EXPRESS listing entities	4550
mapping table	2641
part_template_non_planar_2d_shape	
application assertion	1267, 1288, 1290, 1351, 1357
application object	796
ARM diagrams	4801
ARM EXPRESS listing entities	4550
mapping table	2644
part_template_non_planar_2d_shape_constraint	
AIM EXPRESS expanded listing rules	4190
part_template_or_physical_unit_2d_shape_representation_relationship	
application assertion	1289
application object	798
ARM EXPRESS listing entities	4551
mapping table	2646, 2692
part_template_or_physical_unit_2d_shape_select	
ARM diagrams	4653
ARM EXPRESS listing types	4418
part_template_or_physical_unit_3d_shape_representation_relationship	
application assertion	1289–1290
application object	799
ARM EXPRESS listing entities	4551
mapping table	2649, 2694
part_template_or_physical_unit_3d_shape_select	
ARM diagrams	4653
ARM EXPRESS listing types	4418
part_template_or_physical_unit_shape_select	
ARM EXPRESS listing types	4418
part_template_planar_keepout_shape	
application assertion	1290–1291
application object	802
ARM EXPRESS listing entities	4551
mapping table	2654
part_template_planar_shape	
application assertion	1267, 1289, 1291, 1351, 1357
application object	800

ARM diagrams	4802
ARM EXPRESS listing entities	4551
mapping table	2658
part_template_unique_constraint	
AIM EXPRESS expanded listing rules	4191
part_terminal	
application assertion	1240, 1284
application object	803
ARM diagrams	4798
ARM EXPRESS listing entities	4551
mapping table	2571
part_terminal_external_reference	
application assertion	1176, 1271, 1291
application object	804
ARM diagrams	4677
ARM EXPRESS listing entities	4552
mapping table	1780, 1807
part_terminal_external_reference_unique_constraint	
AIM EXPRESS expanded listing rules	4192
part_text_template	
application assertion	1256, 1291–1292
application object	806
ARM diagrams	4799
ARM EXPRESS listing entities	4552
mapping table	2611
part_text_template_definition	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	3987
AIM EXPRESS short listing entities	3302
mapping table	2611
part_tooling_feature	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3987
AIM EXPRESS short listing entities	3303
application object	807
ARM diagrams	4792
ARM EXPRESS listing entities	4552
mapping table	1887, 2488, 2548, 2571
partially_plated_cutout	
application assertion	1302
application object	808
ARM diagrams	4747
ARM EXPRESS listing entities	4552
mapping table	2162
partially_plated_interconnect_module_edge	
application assertion	1302

application object	809
ARM diagrams	4705
ARM EXPRESS listing entities	4552
mapping table	2162, 2214
partially_routed_physical_network	
application object	810
ARM EXPRESS listing entities	4552
mapping table	2031, 2661
participant_date_and_time_assignment	
application assertion	1283, 1319
application object	811
ARM diagrams	4817
ARM EXPRESS listing entities	4553
mapping table	2796
passage_technology	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3992
AIM EXPRESS short listing entities	3303
application assertion	1202, 1253, 1283, 1292
application object	812
ARM diagrams	4781
ARM EXPRESS listing entities	4553
mapping table	2409, 2416, 2418, 2429
path	
AIM diagrams	4922
AIM EXPRESS expanded listing entities	3992
path_head_to_tail	
AIM EXPRESS expanded listing functions	4285
path_reversed	
AIM EXPRESS expanded listing functions	4285
pca	
application assertion	1292
application object	816
ARM diagrams	4805
ARM EXPRESS listing entities	4553
mapping table	1536
pca_interface_select	
ARM diagrams	4632
ARM EXPRESS listing types	4418
pca_product_design_object_category	
ARM diagrams	4632
ARM EXPRESS listing types	4422
pca_terminal	
application assertion	1293
application object	817
ARM diagrams	4645

ARM EXPRESS listing entities	4553
mapping table	1549
pca_usage_view	
application assertion	1292–1293
application object	818
ARM diagrams	4646
ARM EXPRESS listing entities	4553
mapping table	1550
pcb	
application assertion	1293
application object	818
ARM diagrams	4805
ARM EXPRESS listing entities	4553
mapping table	2163
pcb_usage_view	
application assertion	1293
application object	823
ARM diagrams	4791
ARM EXPRESS listing entities	4554
mapping table	2234
pcurve	
AIM diagrams	4912
AIM EXPRESS expanded listing entities	3992
pcurve_or_surface	
AIM diagrams	4912
AIM EXPRESS expanded listing types	3822
pdf_requires_security_classification	
AIM EXPRESS expanded listing rules	4192
perpendicular_shape_element_relationship	
application assertion	1199
application object	824
ARM diagrams	4723
ARM EXPRESS listing entities	4554
mapping table	1898
perpendicularity_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	3992
AIM EXPRESS short listing entities	3312
application object	825
ARM diagrams	4691
ARM EXPRESS listing entities	4554
mapping table	1898
person	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3993
application assertion	1220, 1293

application object	825
ARM diagrams	4656
ARM EXPRESS listing entities	4554
mapping table	1615
person_and_organization	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3993
application assertion	1193, 1219–1224, 1228, 1276, 1283, 1293, 1322, 1360–1361
application object	826
ARM diagrams	4657
ARM EXPRESS listing entities	4554
mapping table	1615
person_and_organization_assigned_item	
AIM diagrams	4879
AIM EXPRESS expanded listing types	3823
AIM EXPRESS short listing types	2959
person_and_organization_assignment	
AIM diagrams	4879
AIM EXPRESS expanded listing entities	3993
person_and_organization_role	
AIM diagrams	4879
AIM EXPRESS expanded listing entities	3993
AIM EXPRESS short listing entities	3495
person_and_organization_unique_constraint	
AIM EXPRESS expanded listing rules	4193
person_assigned_item	
AIM diagrams	4879
AIM EXPRESS expanded listing types	3823
AIM EXPRESS short listing types	2960
person_assignment	
AIM diagrams	4879
AIM EXPRESS expanded listing entities	3993
person_organization	
ARM diagrams	4655
ARM EXPRESS listing types	4422
person_organization_select	
AIM diagrams	4877
AIM EXPRESS expanded listing types	3823
person_role	
AIM diagrams	4879
AIM EXPRESS expanded listing entities	3993
personal_address	
AIM diagrams	4877
AIM EXPRESS expanded listing entities	3993
mapping table	1615
physical_component	

application assertion	1212, 1243, 1293
application object	827
ARM EXPRESS listing entities	4554
mapping table	1405, 1424, 1438, 1537
physical_component_feature	
application object	828
ARM diagrams	4638
ARM EXPRESS listing entities	4555
mapping table	1898
physical_component_terminal	
application assertion	1251, 1293
application object	828
ARM EXPRESS listing entities	4555
mapping table ...	1479, 1755, 1781, 1793, 1808, 1960, 1977, 1993, 2001, 2006, 2031, 2164, 2215, 2241, 2312, 2360, 2378, 2661, 2729, 2829, 2904
physical_connectivity_abstraction_map	
application assertion	1293
application object	829
ARM diagrams	4738
ARM EXPRESS listing entities	4555
mapping table	2032
physical_connectivity_abstraction_map_unique_constraint	
AIM EXPRESS expanded listing rules	4194
physical_connectivity_definition	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3994
AIM EXPRESS short listing entities	3313
application assertion	1191, 1255, 1293–1296, 1317, 1324, 1327
application object	830
ARM diagrams	4686
ARM EXPRESS listing entities	4555
mapping table	1809, 1814, 2033, 2037
physical_connectivity_definition_relationship	
application assertion	1294
application object	831
ARM diagrams	4687
ARM EXPRESS listing entities	4555
mapping table	1810, 2034
physical_connectivity_definition_relationship_constraint	
AIM EXPRESS expanded listing rules	4195
physical_connectivity_element	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	3994
AIM EXPRESS short listing entities	3314
application assertion	1293–1294, 1316, 1327
application object	832

ARM diagrams	4687
ARM EXPRESS listing entities	4556
mapping table	1811, 2035
physical_connectivity_element_unique_constraint	
AIM EXPRESS expanded listing rules	4196
physical_connectivity_interrupting_cutout	
application assertion	1295
application object	834
ARM diagrams	4747
ARM EXPRESS listing entities	4556
mapping table	2164
physical_connectivity_make_from_relationship	
application assertion	1295
application object	835
ARM diagrams	4620
ARM EXPRESS listing entities	4556
mapping table	1813, 2165
physical_connectivity_structure_definition	
application assertion	1294, 1347
application object	835
ARM diagrams	4686
ARM EXPRESS listing entities	4556
mapping table	1814, 2037
physical_feature_or_part_template	
application assertion	1188, 1217, 1242, 1313, 1318, 1349
application object	836
ARM diagrams	4704
ARM EXPRESS listing entities	4556
mapping table	1899
physical_laminate_component	
application object	837
ARM diagrams	4744
ARM EXPRESS listing entities	4557
mapping table	1899
physical_network	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	3995
AIM EXPRESS short listing entities	3316
application assertion	1192, 1295
application object	837
ARM diagrams	4738
ARM EXPRESS listing entities	4557
mapping table	2031, 2038, 2661
physical_network_supporting_inter_stratum_feature	
application object	840
ARM diagrams	4754

ARM EXPRESS listing entities	4557
mapping table	2041
physical_network_supporting_stratum_feature_conductive_join	
application object	841
ARM diagrams	4737
ARM EXPRESS listing entities	4557
mapping table	2042
physical_node_requirement_to_implementing_component_allocation	
AIM EXPRESS short listing entities	3317
application assertion	1296
application object	842
ARM diagrams	4648
ARM EXPRESS listing entities	4558
mapping table	1479, 2001
physical_unit	
AIM diagrams	4887
AIM EXPRESS expanded listing entities	4000
AIM EXPRESS short listing entities	3319
application assertion	1156, 1274, 1291, 1293, 1296–1299, 1314–1315
application object	844
ARM diagrams	4804
ARM EXPRESS listing entities	4558
mapping table	1503, 1542, 1998–1999, 2115, 2225, 2375, 2670, 2675, 2677, 2763, 2823
physical_unit_3d_keepout_shape	
application assertion	1297
application object	846
ARM EXPRESS listing entities	4558
mapping table	2699
physical_unit_3d_shape	
application assertion	1167, 1174, 1208, 1214, 1267, 1290, 1297–1298, 1335, 1351, 1357
application object	848
ARM diagrams	4807
ARM EXPRESS listing entities	4558
mapping table	2704
physical_unit_datum	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4001
AIM EXPRESS short listing entities	3329
mapping table	1733, 1838, 1847–1848, 1853, 1921
physical_unit_datum_feature	
AIM diagrams	4936
AIM EXPRESS expanded listing entities	4002
AIM EXPRESS short listing entities	3332
mapping table	1850
physical_unit_datum_target	
AIM diagrams	4891

AIM EXPRESS expanded listing entities	4002
AIM EXPRESS short listing entities	3333
mapping table	1859
physical_unit_datum_target_set	
AIM diagrams	4936
AIM EXPRESS expanded listing entities	4002
AIM EXPRESS short listing entities	3334
mapping table	1859
physical_unit_design_view	
application assertion	1155, 1157, 1160, 1298, 1313
application object	850
ARM diagrams	4805
ARM EXPRESS listing entities	4559
mapping table	2675
physical_unit_geometric_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4003
AIM EXPRESS short listing entities	3334
mapping table	1838, 1847, 1873, 1878–1879, 1925
physical_unit_network_definition	
AIM EXPRESS short listing entities	3338
application assertion	1294
application object	852
ARM EXPRESS listing entities	4559
mapping table	1814, 2042, 2763
physical_unit_or_part_template	
ARM diagrams	4677
ARM EXPRESS listing types	4422
physical_unit_or_part_template_or_fp_or_pd	
ARM EXPRESS listing types	4423
physical_unit_planar_keepout_shape	
application assertion	1298–1299
application object	852
ARM EXPRESS listing entities	4559
mapping table	2708
physical_unit_planar_shape	
application assertion	1173, 1209, 1213, 1267, 1289, 1299, 1336, 1351, 1357
application object	855
ARM diagrams	4808
ARM EXPRESS listing entities	4559
mapping table	2714
physical_unit_relationship	
application assertion	1299
application object	857
ARM diagrams	4804
ARM EXPRESS listing entities	4559

mapping table	2675
physical_unit_usage_view	
application assertion	1284–1285, 1298
application object	858
ARM diagrams	4806
ARM EXPRESS listing entities	4560
mapping table	2677
pitch_class	
application assertion	1259, 1299
application object	859
ARM EXPRESS listing entities	4560
mapping table	1651
placement	
AIM diagrams	4905
AIM EXPRESS expanded listing entities	4003
placement_group	
application assertion	1178, 1300
application object	860
ARM diagrams	4630
ARM EXPRESS listing entities	4560
mapping table	1467, 2313
placement_group_area_assignment	
application assertion	1300
application object	862
ARM diagrams	4630
ARM EXPRESS listing entities	4560
mapping table	1468, 2315
placement_group_area_assignment_constraint	
AIM EXPRESS expanded listing rules	4197
placement_group_volume_assignment	
application assertion	1300
application object	862
ARM EXPRESS listing entities	4560
mapping table	1469, 2316
plain_or_structured_text	
ARM diagrams	4669
ARM EXPRESS listing types	4423
planar_curve_constraint	
AIM EXPRESS expanded listing rules	4197
planar_extent	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4003
planar_projected_shape	
application object	863
ARM diagrams	4827
ARM EXPRESS listing entities	4561

mapping table	2850
planar_projected_shape_constraint	
AIM EXPRESS expanded listing rules	4198
planar_purpose	
ARM EXPRESS listing types	4423
planar_shape	
application assertion	1358–1359
application object	864
ARM diagrams	4827
ARM EXPRESS listing entities	4561
mapping table	2850
plane	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	4003
plane_angle_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3823
plane_angle_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4004
mapping table	1635
plane_angle_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4004
mapping table	2836
planned_characteristic	
application assertion	1230, 1300–1301
application object	865
ARM diagrams	4807
ARM EXPRESS listing entities	4561
mapping table	2722
planned_date_effectivity	
application assertion	1301
application object	866
ARM diagrams	4809
ARM EXPRESS listing entities	4561
mapping table	2725
planned_effectivity	
application assertion	1301
application object	867
ARM diagrams	4809
ARM EXPRESS listing entities	4561
mapping table	2725
planned_effectivity_constraint	
AIM EXPRESS expanded listing rules	4198
planned_effectivity_unique_constraint	

AIM EXPRESS expanded listing rules	4199
planned_lot_effectivity	
application assertion	1302
application object	868
ARM diagrams	4809
ARM EXPRESS listing entities	4562
mapping table	2726
planned_serial_number_effectivity	
application object	869
ARM diagrams	4809
ARM EXPRESS listing entities	4562
mapping table	2727
plated_conductive_base_blind_via	
application object	870
ARM diagrams	4739
ARM EXPRESS listing entities	4562
mapping table	2166
plated_cutout	
application object	870
ARM diagrams	4754
ARM EXPRESS listing entities	4562
mapping table	2166
plated_cutout_edge_segment	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	4004
AIM EXPRESS short listing entities	3340
application assertion	1302
application object	871
ARM diagrams	4754
ARM EXPRESS listing entities	4562
mapping table	2166
plated_inter_stratum_feature	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	4005
AIM EXPRESS short listing entities	3341
application object	872
ARM diagrams	4754
ARM EXPRESS listing entities	4562
mapping table	2024, 2041, 2125, 2166–2167, 2215
plated_interconnect_module_edge	
application object	873
ARM diagrams	4754
ARM EXPRESS listing entities	4562
mapping table	2167, 2215
plated_interconnect_module_edge_segment	
AIM diagrams	4892

AIM EXPRESS expanded listing entities	4005
AIM EXPRESS short listing entities	3342
application assertion	1302
application object	875
ARM diagrams	4776
ARM EXPRESS listing entities	4563
mapping table	2167, 2215
plated_passage	
AIM diagrams	4938
AIM EXPRESS expanded listing entities	4006
AIM EXPRESS short listing entities	3344
application object	876
ARM diagrams	4742
ARM EXPRESS listing entities	4563
mapping table	2057–2058, 2078, 2123, 2150, 2166, 2168, 2209
plus_minus_tolerance	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	4007
AIM EXPRESS short listing entities	3495
plus_minus_tolerance_characteristic	
application assertion	1302
application object	877
ARM diagrams	4661
ARM EXPRESS listing entities	4563
mapping table	1652, 1900
plus_minus_tolerance_range_representation	
application assertion	1302
application object	878
ARM diagrams	4719
ARM EXPRESS listing entities	4563
mapping table	1901
plus_minus_tolerance_range_representation_constraint	
AIM EXPRESS expanded listing rules	4199
point	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	4007
point_on_curve	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	4007
point_on_surface	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	4007
point_replica	
AIM diagrams	4904
AIM EXPRESS expanded listing entities	4007
polarity_indication_feature	

application assertion	1302
application object	879
ARM diagrams	4722
ARM EXPRESS listing entities	4563
mapping table	2488, 2571
polarity_indication_feature_constraint	
AIM EXPRESS expanded listing rules	4200
poly_loop	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	4007
polygonal_area	
application assertion	1303
application object	880
ARM EXPRESS listing entities	4563
mapping table	2850
polyline	
AIM diagrams	4911
AIM EXPRESS expanded listing entities	4007
port	11
position_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4009
AIM EXPRESS short listing entities	3346
application object	881
ARM diagrams	4697
ARM EXPRESS listing entities	4563
mapping table	1835, 1838–1839, 1903
position_tolerance_with_referenced_datum_system	
application object	881
ARM diagrams	4691
ARM EXPRESS listing entities	4564
mapping table	1903
position_tolerance_without_referenced_datum_system	
application object	882
ARM diagrams	4696
ARM EXPRESS listing entities	4564
mapping table	1903
positional_boundary	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4010
AIM EXPRESS short listing entities	3352
application assertion	1303
application object	882
ARM diagrams	4712
ARM EXPRESS listing entities	4564
mapping table	1864, 1904, 1912

positional_boundary_and_profile_boundary_member_relationship	
application assertion	1303–1304
application object	883
ARM diagrams	4724
ARM EXPRESS listing entities	4564
mapping table	1904
positional_boundary_definition	
application assertion	1303
application object	885
ARM diagrams	4716
ARM EXPRESS listing entities	4564
mapping table	1904
positional_boundary_member	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4010
AIM EXPRESS short listing entities	3354
application assertion	1217, 1303
application object	885
ARM diagrams	4710
ARM EXPRESS listing entities	4564
mapping table	1905
positional_boundary_member_definition	
application assertion	1217, 1242, 1303
application object	886
ARM diagrams	4721
ARM EXPRESS listing entities	4565
mapping table	1905
positional_boundary_member_definition_constraint	
AIM EXPRESS expanded listing rules	4200
positional_boundary_offset	
application assertion	1304, 1312
application object	888
ARM diagrams	4724
ARM EXPRESS listing entities	4565
mapping table	1906
positional_boundary_offset_constraint	
AIM EXPRESS expanded listing rules	4200
positive_integer	
ARM diagrams	4712
ARM EXPRESS listing types	4423
positive_length_measure	
AIM diagrams	4935
AIM EXPRESS expanded listing types	3823
positive_plane_angle_measure	
AIM diagrams	4936
AIM EXPRESS expanded listing types	3823

positive_ratio_measure	
AIM diagrams	4937
AIM EXPRESS expanded listing types	3823
ARM diagrams	4832
ARM EXPRESS listing types	4423
power_data_element	
application object	889
ARM EXPRESS listing entities	4565
mapping table	1652
power_measure_with_unit	
AIM EXPRESS short listing entities	3355
mapping table	1652
pre_defined_curve_font	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4010
pre_defined_datum_axis_3d_symbol	
application assertion	1304–1305
application object	890
ARM EXPRESS listing entities	4565
mapping table	2851
pre_defined_datum_plane_3d_symbol	
application assertion	1304–1305
application object	891
ARM EXPRESS listing entities	4565
mapping table	2851
pre_defined_dimension_qualifier	
application object	891
ARM diagrams	4715
ARM EXPRESS listing entities	4565
mapping table	1907
pre_defined_geometric_tolerance_qualifier	
application object	892
ARM diagrams	4721
ARM EXPRESS listing entities	4566
mapping table	1908
pre_defined_item	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4010
AIM EXPRESS short listing entities	3496
pre_defined_parallel_datum_axis_2d_symbol	
application assertion	1304
application object	893
ARM EXPRESS listing entities	4566
mapping table	2851
pre_defined_parallel_datum_axis_symbol_3d_2d_relationship	
AIM EXPRESS short listing entities	3356

application assertion	1304
application object	893
ARM EXPRESS listing entities	4566
mapping table	2851
pre_defined_perpendicular_datum_axis_2d_symbol	
application assertion	1305
application object	894
ARM EXPRESS listing entities	4566
mapping table	2852
pre_defined_perpendicular_datum_axis_symbol_3d_2d_relationship	
AIM EXPRESS short listing entities	3356
application assertion	1305
application object	895
ARM EXPRESS listing entities	4566
mapping table	2852
pre_defined_perpendicular_datum_plane_2d_symbol	
application assertion	1305
application object	895
ARM EXPRESS listing entities	4566
mapping table	2853
pre_defined_perpendicular_datum_plane_symbol_3d_2d_relationship	
AIM EXPRESS short listing entities	3357
application assertion	1305
application object	896
ARM EXPRESS listing entities	4566
mapping table	2853
precedence_type	
ARM diagrams	4709
ARM EXPRESS listing types	4423
precision_qualifier	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	4010
application object	897
ARM EXPRESS listing entities	4567
mapping table	1652
predefined_3d_purpose	
ARM EXPRESS listing types	4425
predefined_geometric_status	
ARM EXPRESS listing types	4423
predefined_keepout_purpose	
ARM EXPRESS listing types	4424
predefined_lead_form	
ARM EXPRESS listing types	4424
predefined_planar_purpose	
ARM EXPRESS listing types	4425
preferred_surface_curve_representation	

AIM diagrams	4912
AIM EXPRESS expanded listing types	3823
presentable_text	
AIM diagrams	4933
AIM EXPRESS expanded listing types	3824
presentation_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4011
presentation_representation_select	
AIM diagrams	4866
AIM EXPRESS expanded listing types	3824
presentation_style_assignment	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	4011
presentation_style_by_context	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	4011
presentation_style_select	
AIM diagrams	4931
AIM EXPRESS expanded listing types	3824
pressure_data_element	
application object	897
ARM EXPRESS listing entities	4567
mapping table	1652
pressure_measure_with_unit	
AIM EXPRESS short listing entities	3358
mapping table	1652
primary_orientation_feature	
AIM diagrams	4936
AIM EXPRESS expanded listing entities	4012
AIM EXPRESS short listing entities	3359
application assertion	1306
application object	898
ARM diagrams	4792
ARM EXPRESS listing entities	4567
mapping table	1887, 2488, 2548, 2572
primary_reference_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4012
AIM EXPRESS short listing entities	3360
application object	898
ARM diagrams	4817
ARM EXPRESS listing entities	4567
mapping table	2489, 2572
primary_stratum_indicator_symbol	
application object	901

ARM diagrams	4756
ARM EXPRESS listing entities	4567
mapping table	2169
primitive_2d	
application assertion	1168
application object	903
ARM EXPRESS listing entities	4567
mapping table	2854
printed circuit board layout	11
printed component	11
printed_component	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	4013
AIM EXPRESS short listing entities	3360
application assertion	1306–1307
application object	907
ARM diagrams	4751
ARM EXPRESS listing entities	4568
mapping table	2006, 2169, 2175
printed_component_access_mechanisms	
ARM EXPRESS listing functions	4613
printed_component_join_terminal	
application assertion	1203, 1231, 1260, 1306, 1348
application object	910
ARM diagrams	4756
ARM EXPRESS listing entities	4568
mapping table	2172
printed_component_link	
AIM EXPRESS short listing entities	3363
application assertion	1306–1307
application object	911
ARM EXPRESS listing entities	4568
mapping table	2174
printed_connector_component	
application assertion	1307
application object	912
ARM diagrams	4735
ARM EXPRESS listing entities	4568
mapping table	2006, 2175
printed_connector_component_interface_terminal	
application assertion	1307
application object	913
ARM diagrams	4757
ARM EXPRESS listing entities	4569
mapping table	2176
printed_connector_template	

application assertion	1269, 1307, 1310
application object	914
ARM diagrams	4810
ARM EXPRESS listing entities	4569
mapping table	2729
printed_connector_template_terminal_relationship	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4014
AIM EXPRESS short listing entities	3364
application assertion	1307–1308
application object	916
ARM diagrams	4811
ARM EXPRESS listing entities	4569
mapping table	2729
printed_connector_template_terminal_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4202
printed_part_cross_section_template	
application assertion	1308–1309, 1319
application object	918
ARM diagrams	4810
ARM EXPRESS listing entities	4570
mapping table	2730
printed_part_cross_section_template_terminal	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4014
AIM EXPRESS short listing entities	3366
application assertion	1309
application object	921
ARM diagrams	4812
ARM EXPRESS listing entities	4570
mapping table	2734
printed_part_template	
application assertion	1306, 1309–1311
application object	923
ARM diagrams	4810
ARM EXPRESS listing entities	4570
mapping table	2738
printed_part_template_access_mechanisms	
ARM EXPRESS listing functions	4613
printed_part_template_connected_terminals_definition	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4014
AIM EXPRESS short listing entities	3374
application assertion	1310
application object	925
ARM diagrams	4811

ARM EXPRESS listing entities	4571
mapping table	2740
printed_part_template_interface_terminal	
application assertion	1307–1308, 1310
application object	926
ARM diagrams	4812
ARM EXPRESS listing entities	4571
mapping table	2741
printed_part_template_join_terminal	
application assertion	1306, 1308
application object	927
ARM diagrams	4812
ARM EXPRESS listing entities	4571
mapping table	2742
printed_part_template_link	
AIM EXPRESS short listing entities	3375
application assertion	1307, 1309–1310
application object	928
ARM EXPRESS listing entities	4571
mapping table	2742
printed_part_template_material	
AIM EXPRESS short listing entities	3376
application assertion	1310, 1341
application object	928
ARM diagrams	4789
ARM EXPRESS listing entities	4571
mapping table	2743
printed_part_template_material_link	
AIM EXPRESS short listing entities	3378
application assertion	1309, 1311
application object	930
ARM EXPRESS listing entities	4572
mapping table	2745
printed_part_template_or_printed_part_cross_section_template	
ARM EXPRESS listing types	4425
printed_part_template_terminal	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4015
AIM EXPRESS short listing entities	3380
application assertion	1175, 1241, 1261, 1270, 1310–1311, 1350, 1353
application object	931
ARM diagrams	4812
ARM EXPRESS listing entities	4572
mapping table	2741–2742, 2745
printed_part_template_terminal_class	
ARM diagrams	4812

ARM EXPRESS listing types	4425
printed_part_template_terminal_connection_zone_category	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4015
AIM EXPRESS short listing entities	3382
printed_part_template_terminal_relationship	
application assertion	1311
application object	934
ARM diagrams	4811
ARM EXPRESS listing entities	4572
mapping table	2747
printed_part_template_terminal_relationship_constraint	
AIM EXPRESS expanded listing rules	4203
printed_part_template_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4204
probe_access_area	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	4016
AIM EXPRESS short listing entities	3382
application assertion	1311
application object	935
ARM diagrams	4750
ARM EXPRESS listing entities	4572
mapping table	2123, 2178, 2928–2929
process_specification	
application assertion	1292, 1337, 1341
application object	936
ARM diagrams	4670
ARM EXPRESS listing entities	4573
mapping table	1716
product	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4016
AIM EXPRESS short listing entities	3496
mapping table	1409, 1467, 1759, 1962, 2313, 2760, 2792, 2875
product_association	
application assertion	1311–1312
application object	938
ARM diagrams	4813
ARM EXPRESS listing entities	4573
mapping table	2787
product_association_constraint	
AIM EXPRESS expanded listing rules	4204
product_association_unique_constraint	
AIM EXPRESS expanded listing rules	4205
product_category	

AIM diagrams	4861
AIM EXPRESS expanded listing entities	4016
mapping table	2444–2445, 2666, 2760–2761, 2792
product_category_relationship	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	4017
mapping table	2446, 2668, 2763
product_category_requires_product_related_product_category	
AIM EXPRESS expanded listing rules	4206
product_concept	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	4017
AIM EXPRESS short listing entities	3496
mapping table	1719
product_concept_context	
AIM diagrams	4859
AIM EXPRESS expanded listing entities	4017
product_concept_requires_configuration_item	
AIM EXPRESS expanded listing rules	4206
product_context	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4017
product_definition	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4017
AIM EXPRESS short listing entities	3497
mapping table ...	1555, 1742, 1749, 1760, 1764, 1773, 1782, 1787, 1793, 1797, 1802, 1818, 2097, 2189, 2448, 2617, 2763, 2766, 2877, 2880
product_definition_context	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4017
mapping table	2765
product_definition_effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	4017
product_definition_formation	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4018
AIM EXPRESS short listing entities	3497
mapping table	2784, 2796
product_definition_formation_relationship	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	4018
AIM EXPRESS short listing entities	3498
mapping table	2786–2787
product_definition_formation_requires_approval	

AIM EXPRESS expanded listing rules	4206
product_definition_formation_requires_person_organization	
AIM EXPRESS expanded listing rules	4207
product_definition_formation_with_specified_source	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4018
mapping table	2874
product_definition_relationship	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4018
AIM EXPRESS short listing entities	3498
mapping table ... 1446, 1560–1562, 1564, 1570–1571, 1694, 1743, 1745–1746, 1750, 1752–1753,	
1757, 1772, 1775, 1777, 1786, 1789, 1791–1792, 1816, 2110, 2236, 2238–2239, 2241, 2249, 2257,	
2260, 2531, 2675, 2788	
product_definition_requires_date_or_date_and_time	
AIM EXPRESS expanded listing rules	4207
product_definition_requires_person_organization	
AIM EXPRESS expanded listing rules	4207
product_definition_shape	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	4018
product_definition_usage	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4019
AIM EXPRESS short listing entities	3500
product_definition_with_associated_documents	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4019
AIM EXPRESS short listing entities	3500
mapping table	1555, 2763
product_material_composition_relationship	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	4019
AIM EXPRESS short listing entities	3501
mapping table	1958, 2447
product_object_select	
ARM diagrams	4649
ARM EXPRESS listing types	4426
product_related_characterized_product_category	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3384
mapping table	2447, 2668, 2782
product_related_product_category	
AIM diagrams	4861
AIM EXPRESS expanded listing entities	4020

product_requires_person_organization	
AIM EXPRESS expanded listing rules	4207
product_requires_product_definition_formation	
AIM EXPRESS expanded listing rules	4208
product_specific_parameter_value_assignment	
AIM EXPRESS short listing entities	3384
mapping table	2668, 2782
profile_boundary_definition	
application assertion	1312
application object	939
ARM diagrams	4700
ARM EXPRESS listing entities	4573
mapping table	1908
profile_boundary_definition_with_offsets	
application assertion	1312
application object	940
ARM diagrams	4698
ARM EXPRESS listing entities	4573
mapping table	1909
profile_boundary_definition_with_offsets_constraint	
AIM EXPRESS expanded listing rules	4208
profile_of_any_line_tolerance	
application object	941
ARM diagrams	4693
ARM EXPRESS listing entities	4573
mapping table	1910
profile_of_any_line_tolerance_with_referenced_datum_system	
application object	942
ARM diagrams	4691
ARM EXPRESS listing entities	4573
mapping table	1910
profile_of_any_line_tolerance_without_referenced_datum_system	
application object	943
ARM diagrams	4696
ARM EXPRESS listing entities	4574
mapping table	1911
profile_of_any_surface_tolerance	
application object	944
ARM diagrams	4696
ARM EXPRESS listing entities	4574
mapping table	1911
profile_of_any_surface_tolerance_with_referenced_datum_system	
application assertion	1168
application object	945
ARM diagrams	4691
ARM EXPRESS listing entities	4574

mapping table	1911
profile_of_any_surface_tolerance_without_referenced_datum_system	
application object	945
ARM diagrams	4696
ARM EXPRESS listing entities	4574
mapping table	1912
profile_related_positional_boundary	
application assertion	1303, 1312
application object	946
ARM diagrams	4712
ARM EXPRESS listing entities	4574
mapping table	1912
profile_related_positional_boundary_definition	
application assertion	1242, 1312
application object	947
ARM diagrams	4716
ARM EXPRESS listing entities	4574
mapping table	1912
profile_related_positional_boundary_definition_constraint	
AIM EXPRESS expanded listing rules	4208
projected_tolerance_zone_definition	
application assertion	1313
application object	948
ARM diagrams	4701
ARM EXPRESS listing entities	4574
mapping table	1913
projected_zone_and_base_relationship	
application assertion	1313
application object	949
ARM diagrams	4724
ARM EXPRESS listing entities	4575
mapping table	1913
projected_zone_and_base_relationship_constraint	
AIM EXPRESS expanded listing rules	4209
projected_zone_definition	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3501
mapping table	1913
projected_zone_height_characteristic	
application assertion	1313
application object	950
ARM diagrams	4725
ARM EXPRESS listing entities	4575
mapping table	1913
projected_zone_height_characteristic_constraint	

AIM EXPRESS expanded listing rules	4209
promissory_usage_in_product_model	
AIM EXPRESS short listing entities	3385
promissory_usage_in_product_model_assigned_item	
AIM EXPRESS short listing types	2960
promissory_usage_in_product_model_relationship	
application object	952
ARM EXPRESS listing entities	4575
mapping table	1406, 1960, 2179, 2378, 2829
promissory_usage_occurrence	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3501
mapping table	1408, 1962, 2181, 2380, 2831
promissory_usage_relationship	
application assertion	1313–1314
application object	953
ARM diagrams	4622
ARM EXPRESS listing entities	4575
mapping table	1408, 1962, 2181, 2380, 2831
promissory_usage_relationship_constraint	
AIM EXPRESS expanded listing rules	4209
promissory_usage_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4210
property_definition	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3502
mapping table ...	1837, 1845–1846, 1848–1851, 1854, 1858–1859, 1864–1865, 1870, 1904–1906, 1908–1909, 1912, 1914, 1927, 1929–1930, 1932, 1934–1935, 1938–1939, 2256, 2811, 2868, 2870
property_definition_relationship	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3504
mapping table	1765–1766, 1768, 1773, 1787, 1998, 2816
property_definition_representation	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	4020
AIM EXPRESS short listing entities	3505
mapping table	1656, 1825, 1832, 2232, 2351, 2365, 2510, 2515, 2534–2535, 2586, 2687–2688
purpose	
ARM diagrams	4807
purpose_occurrence	
application object	954
ARM diagrams	4820
ARM EXPRESS listing entities	4575

mapping table	2816
qualified_data_element	
ARM diagrams	4659
qualitative_uncertainty	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	4021
application object	954
ARM EXPRESS listing entities	4575
mapping table	1652
quantified_assembly_component_usage	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4021
mapping table	1403–1404, 1501, 1532, 1955, 1959, 2149, 2377, 2822, 2828
quasi_uniform_curve	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	4021
quasi_uniform_surface	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	4021
radius_edge_feature_shape	
application assertion	1314
application object	955
ARM diagrams	4674
ARM EXPRESS listing entities	4575
mapping table	1737
radius_edge_feature_shape_constraint	
AIM EXPRESS expanded listing rules	4211
range_characteristic	
application assertion	1266, 1299, 1314
application object	955
ARM EXPRESS listing entities	4576
mapping table	1652
range_class	
ARM EXPRESS listing types	4426
ratio_data_element	
application object	957
ARM EXPRESS listing entities	4576
mapping table	1653
ratio_measure	
AIM diagrams	4927
AIM EXPRESS expanded listing types	3824
ratio_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4021
mapping table	1646, 1653, 2429
ratio_unit	

AIM diagrams	4894
AIM EXPRESS expanded listing entities	4021
rational_b_spline_curve	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	4022
rational_b_spline_surface	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	4022
rectangular_area	
application assertion	1314
application object	958
ARM EXPRESS listing entities	4576
mapping table	2854
rectangular_array_placement_group_component	
application assertion	1314
application object	958
ARM EXPRESS listing entities	4576
mapping table	1439, 1978
rectangular_array_placement_group_component_definition	
AIM EXPRESS short listing entities	3386
mapping table	1439, 1978
rectangular_array_placement_group_component_shape_aspect	
AIM EXPRESS short listing entities	3386
mapping table	1978
rectangular_composite_array_shape_aspect	
AIM EXPRESS short listing entities	3387
mapping table	2567
rectangular_composite_surface	
AIM diagrams	4909
AIM EXPRESS expanded listing entities	4022
rectangular_trimmed_surface	
AIM diagrams	4909
AIM EXPRESS expanded listing entities	4023
reference_composition_path	
application assertion	1205, 1315
application object	959
ARM diagrams	4685
ARM EXPRESS listing entities	4576
mapping table	1782, 1793, 1797, 1802
reference_composition_path_constraint	
AIM EXPRESS expanded listing rules	4211
reference_functional_unit_assignment_to_part	
application assertion	1315
application object	960
ARM diagrams	4841
ARM EXPRESS listing entities	4576

mapping table	1783, 1795, 1799, 1804
reference_functional_unit_assignment_to_part_constraint	
AIM EXPRESS expanded listing rules	4212
reference_functional_unit_assignment_to_part_unique_constraint	
AIM EXPRESS expanded listing rules	4213
reference_packaged_part_assembly_implementation	
AIM EXPRESS short listing entities	3387
application object	962
ARM EXPRESS listing entities	4577
mapping table	1608
reference_packaged_part_interconnect_implementation	
AIM EXPRESS short listing entities	3388
application assertion	1235
application object	962
ARM EXPRESS listing entities	4577
mapping table	1608
registered_font	
application assertion	1291, 1315, 1339
application object	963
ARM diagrams	4838
ARM EXPRESS listing entities	4577
mapping table	2926
registered_font_constraint	
AIM EXPRESS expanded listing rules	4213
registered_font_unique_constraint	
AIM EXPRESS expanded listing rules	4214
registered_style	
application assertion	1277, 1315
ARM diagrams	4792
registered_style_constraint	
AIM EXPRESS expanded listing rules	4214
registered_style_unique_constraint	
AIM EXPRESS expanded listing rules	4215
relationship_operation	
ARM diagrams	4822
ARM EXPRESS listing types	4426
reparametrised_composite_curve_segment	
AIM diagrams	4917
AIM EXPRESS expanded listing entities	4023
replacement_from_item	
AIM diagrams	4871
AIM EXPRESS expanded listing types	3824
AIM EXPRESS short listing types	2961
replacement_to_item	
AIM diagrams	4871
AIM EXPRESS expanded listing types	3824

AIM EXPRESS short listing types	2961
representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4023
AIM EXPRESS short listing entities	3505
ARM diagrams	4833
mapping table	1390, 1399, 1446, 1488, 1518–1519, 1551, 1643, 1651, 1659, 1667, 1704, 1779–1780, 1807, 1821, 1825, 1836, 1860, 1868, 1886, 2010, 2062, 2259, 2261, 2586, 2687, 2722, 2793–2796, 2799, 2921
representation_context	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4023
AIM EXPRESS short listing entities	3507
representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	4023
AIM EXPRESS short listing entities	3507
mapping table ...	1395, 1486, 1516, 1635–1636, 1648, 1675, 1702, 1720, 1834, 2008, 2373, 2428, 2712, 2717
representation_map	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	4024
representation_relationship	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4024
AIM EXPRESS short listing entities	3507
mapping table	1397, 1677, 2364, 2720
representation_relationship_with_transformation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4024
mapping table	1594, 1597, 2353, 2357, 2847, 2854, 2863, 2865
requirement_allocation	
application assertion	1300, 1316–1318
application object	964
ARM diagrams	4649
ARM EXPRESS listing entities	4577
mapping table	1564, 2241, 2788
requirement_allocation_group	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4024
AIM EXPRESS short listing entities	3390
mapping table	1481, 1564, 2003, 2241, 2788
requirement_allocation_select	
ARM diagrams	4807
ARM EXPRESS listing types	4426
requirement_composition	

application assertion	1318
application object	965
ARM diagrams	4822
ARM EXPRESS listing entities	4577
mapping table	2816
requirement_composition_constraint	
AIM EXPRESS expanded listing rules	4215
requirement_definition	
AIM EXPRESS short listing entities	3391
application assertion	1226
application object	966
ARM EXPRESS listing entities	4577
mapping table	2763, 2817
requirement_group_occurrence	
application assertion	1318
application object	967
ARM diagrams	4820
ARM EXPRESS listing entities	4578
mapping table	2817
requirements_property	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4026
AIM EXPRESS short listing entities	3393
mapping table	1554, 2239–2240, 2804, 2806, 2811–2816, 2818, 2821
requirements_property_group	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4026
AIM EXPRESS short listing entities	3396
mapping table	2817
resistance_data_element	
application object	967
ARM EXPRESS listing entities	4578
mapping table	1653
resistance_measure_with_unit	
AIM EXPRESS short listing entities	3396
mapping table	1653
restrained_item_select	
ARM diagrams	4725
ARM EXPRESS listing types	4427
restraint_condition	
application assertion	1250, 1318–1319
application object	968
ARM diagrams	4725
ARM EXPRESS listing entities	4578
mapping table	1914
restraint_condition_constraint	

AIM EXPRESS expanded listing rules	4216
restrict_action_request_status	
AIM EXPRESS expanded listing rules	4216
restrict_approval_status	
AIM EXPRESS expanded listing rules	4216
restrict_date_role	
AIM EXPRESS expanded listing rules	4216
restrict_manifold_surface_shape_representation	
AIM EXPRESS expanded listing rules	4217
restriction_basis_item	
ARM diagrams	4629
ARM EXPRESS listing types	4427
reversible_topology	
AIM diagrams	4922
AIM EXPRESS expanded listing types	3824
reversible_topology_item	
AIM diagrams	4922
AIM EXPRESS expanded listing types	3824
revolved_area_solid	
AIM diagrams	4925
AIM EXPRESS expanded listing entities	4026
revolved_face_solid	
AIM diagrams	4935
AIM EXPRESS expanded listing entities	4026
right_angular_wedge	
AIM diagrams	4924
AIM EXPRESS expanded listing entities	4027
application object	969
ARM diagrams	4690
ARM EXPRESS listing entities	4578
mapping table	1658
right_circular_cone	
AIM diagrams	4925
AIM EXPRESS expanded listing entities	4027
right_circular_cylinder	
AIM diagrams	4927
AIM EXPRESS expanded listing entities	4027
application object	969
ARM diagrams	4690
ARM EXPRESS listing entities	4578
mapping table	1658
rigid printed circuit board	11
rigid-flex printed circuit board	11
routed_interconnect_component	
application assertion	1319
application object	969

ARM diagrams	4626
ARM EXPRESS listing entities	4578
mapping table	1425
routed_packaged_component	
ARM diagrams	4626
routed_physical_component	
AIM EXPRESS short listing entities	3397
application assertion	1319
application object	970
ARM EXPRESS listing entities	4578
mapping table	1538
routed_printed_component	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	4028
AIM EXPRESS short listing entities	3398
application assertion	1319, 1348
application object	971
ARM diagrams	4744
ARM EXPRESS listing entities	4578
mapping table	2182
rule_action	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	4029
AIM EXPRESS short listing entities	3400
application assertion	1319–1320
application object	972
ARM diagrams	4818
ARM EXPRESS listing entities	4579
mapping table	2797–2802
rule_action_unique_constraint	
AIM EXPRESS expanded listing rules	4218
rule_adoption	
application object	973
ARM diagrams	4818
ARM EXPRESS listing entities	4579
mapping table	2798
rule_based_constraint_occurrence	
application assertion	1320
application object	974
ARM EXPRESS listing entities	4579
mapping table	2818
rule_based_evaluated_characteristic	
application object	974
ARM EXPRESS listing entities	4579
mapping table	2723
rule_change_request	

application assertion	1320
application object	975
ARM diagrams	4818
ARM EXPRESS listing entities	4579
mapping table	2798
rule_complex_clause	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4029
AIM EXPRESS short listing entities	3403
mapping table	2793
rule_conclusion_definition	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4029
AIM EXPRESS short listing entities	3403
mapping table	2793
rule_creation	
application object	975
ARM diagrams	4818
ARM EXPRESS listing entities	4579
mapping table	2798
rule_definition	
AIM diagrams	4863
AIM EXPRESS expanded listing entities	4030
AIM EXPRESS short listing entities	3404
mapping table	2763, 2794
rule_expiration	
application object	976
ARM diagrams	4818
ARM EXPRESS listing entities	4579
mapping table	2798
rule_function_definition	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4030
AIM EXPRESS short listing entities	3405
mapping table	2794
rule_justification	
application assertion	1320
application object	976
ARM diagrams	4818
ARM EXPRESS listing entities	4580
mapping table	2798
rule_modification	
application assertion	1320
application object	977
ARM diagrams	4818
ARM EXPRESS listing entities	4580

mapping table	2799
rule_premise_definition	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4031
AIM EXPRESS short listing entities	3406
mapping table	2795
rule_priority	
application assertion	1320
application object	978
ARM diagrams	4819
ARM EXPRESS listing entities	4580
mapping table	2799
rule_priority_constraint	
AIM EXPRESS expanded listing rules	4218
rule_rejection	
application object	979
ARM diagrams	4818
ARM EXPRESS listing entities	4580
mapping table	2800
rule_replacement_from_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	4031
AIM EXPRESS short listing entities	3407
rule_replacement_from_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	4031
rule_replacement_to_assignment	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	4031
AIM EXPRESS short listing entities	3407
rule_replacement_to_request_assignment	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	4032
rule_request	
application object	979
ARM diagrams	4818
ARM EXPRESS listing entities	4580
mapping table	2800
rule_set	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4032
AIM EXPRESS short listing entities	3408
application assertion	1320
application object	980
ARM diagrams	4819
ARM EXPRESS listing entities	4580

mapping table	2800–2801
rule_set_group	
application assertion	1320
application object	981
ARM diagrams	4819
ARM EXPRESS listing entities	4580
mapping table	2801
rule_simple_clause	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4032
AIM EXPRESS short listing entities	3409
mapping table	2796
rule_submission	
application object	981
ARM diagrams	4818
ARM EXPRESS listing entities	4581
mapping table	2801
rule_supersedence	
application assertion	1320
application object	982
ARM diagrams	4818
ARM EXPRESS listing entities	4581
mapping table	2802
runout_tolerance_zone_definition_with_specified_angle	
application assertion	1320
application object	983
ARM diagrams	4701
ARM EXPRESS listing entities	4581
mapping table	1915
runout_tolerance_zone_orienting_relationship	
application assertion	1321
application object	984
ARM diagrams	4726
ARM EXPRESS listing entities	4581
mapping table	1915
runout_zone_definition	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4033
AIM EXPRESS short listing entities	3508
mapping table	1915
runout_zone_orientation	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4033
AIM EXPRESS short listing entities	3508
application assertion	1320–1321
application object	985

ARM diagrams	4726
ARM EXPRESS listing entities	4581
mapping table	1915
runout_zone_orientation_reference_direction	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4033
AIM EXPRESS short listing entities	3508
mapping table	1915
same_stratum	
ARM EXPRESS listing functions	4614
scalar_or_coordinated_characteristics	
ARM diagrams	4618
ARM EXPRESS listing types	4427
scalar_terminal_definition	
application assertion	1237, 1240–1241, 1321
application object	986
ARM diagrams	4683
ARM EXPRESS listing entities	4581
mapping table	1830
scalar_terminal_definition_link	
AIM EXPRESS short listing entities	3410
application assertion	1321
application object	986
ARM EXPRESS listing entities	4582
mapping table	1830
scalar_times_vector	
AIM EXPRESS expanded listing functions	4286
scattering_parameter_data_element	
application object	987
ARM EXPRESS listing entities	4582
mapping table	1653
scattering_parameter_measure_with_unit	
AIM EXPRESS short listing entities	3412
mapping table	1653
seam_curve	
AIM diagrams	4912
AIM EXPRESS expanded listing entities	4033
seating_plane	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4033
AIM EXPRESS short listing entities	3412
application assertion	1164, 1190, 1277–1278
application object	988
ARM diagrams	4710
ARM EXPRESS listing entities	4582
mapping table	1632, 2489

seating_plane_intersection_type	
ARM EXPRESS listing types	4427
second_in_minute	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3825
second_proj_axis	
AIM EXPRESS expanded listing functions	4286
secondary_orientation_feature	
AIM diagrams	4936
AIM EXPRESS expanded listing entities	4034
AIM EXPRESS short listing entities	3413
application assertion	1321
application object	989
ARM diagrams	4695
ARM EXPRESS listing entities	4582
mapping table	1887, 2490, 2548, 2572
security_classification	
AIM diagrams	4874
AIM EXPRESS expanded listing entities	4034
AIM EXPRESS short listing entities	3509
application assertion	1157, 1160, 1220, 1223, 1225, 1313, 1321–1322
application object	989
ARM diagrams	4658
ARM EXPRESS listing entities	4582
mapping table	1615
security_classification_assigned_item	
AIM diagrams	4874
AIM EXPRESS expanded listing types	3825
AIM EXPRESS short listing types	2961
security_classification_assignment	
AIM diagrams	4874
AIM EXPRESS expanded listing entities	4034
security_classification_constraint	
AIM EXPRESS expanded listing rules	4219
security_classification_level	
AIM diagrams	4874
AIM EXPRESS expanded listing entities	4034
AIM EXPRESS short listing entities	3509
mapping table	1615
separate_geometric_tolerance_requirement	
application assertion	1322
application object	991
ARM diagrams	4726
ARM EXPRESS listing entities	4582
mapping table	1916
sequential_laminate_passage_based_fabrication_joint	

AIM diagrams	4891
AIM EXPRESS expanded listing entities	4034
AIM EXPRESS short listing entities	3414
mapping table	1728, 2827
serial_numbered_effectivity	
AIM diagrams	4864
AIM EXPRESS expanded listing entities	4035
mapping table	2727
set_characteristic	
application assertion	1171, 1322
ARM EXPRESS listing types	4427
set_of_reversible_topology_item	
AIM diagrams	4922
AIM EXPRESS expanded listing types	3825
set_of_topology_reversed	
AIM EXPRESS expanded listing functions	4287
shape_and_ee_product_definition_based_constraint_occurrence	
application assertion	1322
application object	992
ARM EXPRESS listing entities	4582
mapping table	2818
shape_aspect	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4035
AIM EXPRESS short listing entities	3509
mapping table ...	1476, 1496, 1517, 1525, 1603, 1736, 1770, 1809, 1811, 1814, 1827, 1829, 1884, 1887, 1899, 1916–1917, 1926, 1929, 1942–1944, 1949–1950, 2000, 2028, 2033, 2035, 2038, 2044, 2083, 2134–2135, 2141, 2155, 2193, 2217, 2219, 2221–2222, 2371, 2374, 2427, 2430, 2488, 2506, 2527, 2532, 2542, 2544, 2546, 2548, 2571, 2661, 2740, 2747, 2839, 2879, 2905
shape_aspect_deriving_relationship	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	4035
AIM EXPRESS short listing entities	3511
mapping table	1918
shape_aspect_relationship	
AIM diagrams	4893
AIM EXPRESS expanded listing entities	4035
AIM EXPRESS short listing entities	3511
mapping table ...	1444, 1468–1471, 1473–1474, 1496, 1560–1561, 1564, 1629–1630, 1756, 1778, 1783, 1792, 1795, 1797, 1799, 1802, 1804, 1810, 1845, 1848, 1851–1852, 1859, 1866, 1886–1887, 1898, 1904, 1915, 1918–1919, 1929, 1986, 1988–1989, 1991, 1993, 2027, 2032, 2034, 2046, 2059–2060, 2070, 2124, 2205–2206, 2224, 2236, 2238, 2241, 2260, 2285, 2315–2316, 2371, 2427, 2456, 2487, 2514, 2522, 2540, 2547, 2747, 2788, 2901, 2910
shape_composition_relationship	
application assertion	1322
application object	993

ARM EXPRESS listing entities	4583
mapping table	2854
shape_definition	
AIM diagrams	4865
AIM EXPRESS expanded listing types	3825
application assertion	1241, 1322–1323, 1350
application object	994
ARM EXPRESS listing entities	4583
mapping table	2856
shape_definition_representation	
AIM diagrams	4868
AIM EXPRESS expanded listing entities	4035
shape_dimension_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	4036
AIM EXPRESS short listing entities	3514
application assertion	1166, 1216
application object	995
ARM diagrams	4719
ARM EXPRESS listing entities	4583
mapping table	1872, 1882–1883, 1901, 1916, 1921, 1927
shape_dimension_representation_constraint	
AIM EXPRESS expanded listing rules	4220
shape_element	
application assertion	1216, 1218, 1323
application object	996
ARM diagrams	4727
ARM EXPRESS listing entities	4583
mapping table	1916
shape_element_boundary_relationship	
application object	997
ARM diagrams	4723
ARM EXPRESS listing entities	4584
mapping table	1918
shape_element_composing_relationship	
application assertion	1323
application object	998
ARM diagrams	4723
ARM EXPRESS listing entities	4584
mapping table	1918
shape_element_composing_relationship_constraint	
AIM EXPRESS expanded listing rules	4220
shape_element_constituent_relationship	
application object	999
ARM diagrams	4723
ARM EXPRESS listing entities	4584

mapping table	1918
shape_element_constituent_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4221
shape_element_constraint	
AIM EXPRESS expanded listing rules	4222
shape_element_deriving_relationship	
application assertion	1323
application object	1000
ARM diagrams	4723
ARM EXPRESS listing entities	4584
mapping table	1918
shape_element_deriving_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4223
shape_element_locating_relationship	
application assertion	1216, 1323
application object	1001
ARM diagrams	4727
ARM EXPRESS listing entities	4584
mapping table	1919
shape_element_locating_relationship_unique_constraint	
AIM EXPRESS expanded listing rules	4224
shape_element_relationship	
application assertion	1323
application object	1002
ARM diagrams	4723
ARM EXPRESS listing entities	4584
mapping table	1919
shape_element_unique_constraint	
AIM EXPRESS expanded listing rules	4225
shape_formed_terminal	
AIM EXPRESS short listing entities	3415
application assertion	1323
application object	1003
ARM diagrams	4791
ARM EXPRESS listing entities	4585
mapping table	2491
shape_item	
application assertion	1323
application object	1004
ARM EXPRESS listing entities	4585
mapping table	2857
shape_location	
ARM EXPRESS listing types	4427
shape_modification	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4037

AIM EXPRESS short listing entities	3415
mapping table	2890, 2902–2904, 2909
shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	4037
AIM EXPRESS short listing entities	3514
mapping table	1556, 1573–1574, 1607, 1729, 1735, 1737–1738, 2318–2319, 2348, 2351, 2360, 2362–2363, 2367, 2628, 2630, 2635, 2639, 2654, 2658, 2699, 2704, 2708, 2714, 2835, 2841, 2849–2853, 2856–2857
shape_representation_relationship	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4037
AIM EXPRESS short listing entities	3515
mapping table	1588, 1591, 2339, 2344, 2646, 2649, 2692, 2694, 2847
shape_representation_relationship_with_transformation	
AIM EXPRESS short listing entities	3418
mapping table	2342
shape_representation_requires_origin	
AIM EXPRESS expanded listing rules	4225
shell	
AIM diagrams	4923
AIM EXPRESS expanded listing types	3825
shell_based_2d_wireframe	
application object	1005
ARM diagrams	4827
ARM EXPRESS listing entities	4585
mapping table	2931
shell_based_2d_wireframe_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	4041
AIM EXPRESS short listing entities	3419
mapping table	2931
shell_based_surface_model	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	4041
shell_based_wireframe	
application object	1005
ARM diagrams	4823
ARM EXPRESS listing entities	4585
mapping table	2932
shell_based_wireframe_model	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	4041
shell_based_wireframe_shape_representation	
AIM diagrams	4901
AIM EXPRESS expanded listing entities	4045

mapping table	2932
shell_reversed	
AIM EXPRESS expanded listing functions	4288
shield	
application assertion	1324
application object	1006
ARM diagrams	4625
ARM EXPRESS listing entities	4585
mapping table	1426, 2185
shield_allocation	
application assertion	1324
application object	1008
ARM diagrams	4647
ARM EXPRESS listing entities	4585
mapping table	1570, 2249
shieldable_item	
ARM diagrams	4625
ARM EXPRESS listing types	4428
si_prefix	
AIM diagrams	4894
AIM EXPRESS expanded listing types	3826
ARM EXPRESS listing types	4428
si_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4045
mapping table	1645, 2837
si_unit_name	
AIM diagrams	4894
AIM EXPRESS expanded listing types	3826
signal	
AIM diagrams	4865
AIM EXPRESS expanded listing entities	4045
AIM EXPRESS short listing entities	3425
application assertion	1324
application object	1008
ARM diagrams	4833
ARM EXPRESS listing entities	4586
mapping table	2869
signal_category	
AIM diagrams	4884
AIM EXPRESS expanded listing entities	4045
AIM EXPRESS short listing entities	3426
signal_flow_direction	
ARM diagrams	4617
ARM EXPRESS listing types	4428
signal_property	

application assertion	1185, 1324–1325
application object	1009
ARM diagrams	4833
ARM EXPRESS listing entities	4586
mapping table	2870
signal_specification	
application assertion	1236, 1325
application object	1011
ARM diagrams	4688
ARM EXPRESS listing entities	4586
mapping table	1821
simultaneous_geometric_tolerance_requirement	
application assertion	1325
application object	1013
ARM diagrams	4727
ARM EXPRESS listing entities	4586
mapping table	1921
single-sided printed circuit board	11
single_boundary_dimension	
application object	1014
ARM diagrams	4702
ARM EXPRESS listing entities	4586
mapping table	1921
single_datum	
application assertion	1198
application object	1015
ARM diagrams	4673
ARM EXPRESS listing entities	4586
mapping table	1921
single_package_version	
ARM EXPRESS listing functions	4613
singular_dimension_representation	
application assertion	1326
application object	1015
ARM diagrams	4718
ARM EXPRESS listing entities	4587
mapping table	1921
size_select	
AIM diagrams	4928
AIM EXPRESS expanded listing types	3826
solid_angle_data_element	
application object	1016
ARM EXPRESS listing entities	4587
mapping table	1653
solid_angle_measure	
AIM diagrams	4896

AIM EXPRESS expanded listing types	3826
solid_angle_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4046
mapping table	1653
solid_angle_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4046
mapping table	2836
solid_curve	
application assertion	1194
application object	1017
ARM diagrams	4838
ARM EXPRESS listing entities	4587
mapping table	2925
solid_curve_font	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4046
AIM EXPRESS short listing entities	3426
mapping table	2925
solid_fill_style	
application assertion	1233
application object	1017
ARM diagrams	4832
ARM EXPRESS listing entities	4587
mapping table	2857
solid_model	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	4046
solid_of_linear_extrusion	
application object	1017
ARM diagrams	4823
ARM EXPRESS listing entities	4587
mapping table	2872
solid_replica	
AIM diagrams	4926
AIM EXPRESS expanded listing entities	4046
source	
AIM diagrams	4860
AIM EXPRESS expanded listing types	3827
source_item	
AIM diagrams	4886
AIM EXPRESS expanded listing types	3827
spacing_type	
ARM EXPRESS listing types	4429
special_symbol_laminate_component	

application assertion	1175, 1326
application object	1018
ARM diagrams	4756
ARM EXPRESS listing entities	4587
mapping table	2187
special_symbol_part_template	
application assertion	1326
application object	1019
ARM diagrams	4755
ARM EXPRESS listing entities	4587
mapping table	2615
specification_allocation	
application assertion	1326–1329
application object	1019
ARM diagrams	4763
ARM EXPRESS listing entities	4587
mapping table	2250
specification_product_object_select	
ARM EXPRESS listing types	4429
specified_higher_usage_occurrence	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4047
AIM EXPRESS short listing entities	3516
mapping table	1558
sphere	
AIM diagrams	4925
AIM EXPRESS expanded listing entities	4047
spherical_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	4047
square_or_round	
ARM diagrams	4837
ARM EXPRESS listing types	4429
squared_or_rounded	
AIM diagrams	4928
AIM EXPRESS expanded listing types	3827
stackup_select	
ARM EXPRESS listing types	4429
standard_document_identification_string	
ARM diagrams	4618
ARM EXPRESS listing types	4430
standard_or_reference_document	
ARM diagrams	4618
ARM EXPRESS listing types	4430
standard_uncertainty	
AIM diagrams	4889

AIM EXPRESS expanded listing entities	4047
application object	1021
ARM EXPRESS listing entities	4587
mapping table	1653
start_order	
application object	1021
ARM diagrams	4664
ARM EXPRESS listing entities	4588
mapping table	1680
start_request	
AIM diagrams	4870
AIM EXPRESS expanded listing entities	4047
AIM EXPRESS short listing entities	3427
application object	1021
ARM diagrams	4665
ARM EXPRESS listing entities	4588
mapping table	1680
start_request_item	
AIM diagrams	4870
AIM EXPRESS expanded listing types	3827
AIM EXPRESS short listing types	2962
start_work	
AIM diagrams	4871
AIM EXPRESS expanded listing entities	4047
AIM EXPRESS short listing entities	3427
statistical_dimensional_location	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	4048
AIM EXPRESS short listing entities	3428
mapping table	1922
statistical_dimensional_size	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	4049
AIM EXPRESS short listing entities	3428
mapping table	1922
statistical_dimensional_tolerance	
application assertion	1329
application object	1022
ARM diagrams	4717
ARM EXPRESS listing entities	4588
mapping table	1922
statistical_geometric_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4049
AIM EXPRESS short listing entities	3430
application assertion	1329

application object	1023
ARM diagrams	4720
ARM EXPRESS listing entities	4588
mapping table	1924
statistical_tolerance_characteristic	
application object	1024
ARM diagrams	4661
ARM EXPRESS listing entities	4588
mapping table	1653
status	
ARM diagrams	4765
ARM EXPRESS listing types	4430
step_edge_feature_shape	
application assertion	1329
application object	1025
ARM diagrams	4674
ARM EXPRESS listing entities	4588
mapping table	1738
step_edge_feature_shape_constraint	
AIM EXPRESS expanded listing rules	4226
stiffness_class	
ARM diagrams	4783
ARM EXPRESS listing types	4430
straightness_tolerance	
application object	1027
ARM diagrams	4696
ARM EXPRESS listing entities	4588
mapping table	1925
stratum	
AIM diagrams	4860
AIM EXPRESS expanded listing entities	4051
AIM EXPRESS short listing entities	3432
application assertion ... 1178, 1206, 1248, 1252, 1254, 1258, 1274, 1317, 1328–1331, 1336–1337,	
1341	
application object	1028
ARM diagrams	4748
ARM EXPRESS listing entities	4589
mapping table	2096, 2099, 2189, 2763, 2877–2878, 2880
stratum_3d_shape	
application assertion	1330, 1332, 1336–1337
application object	1031
ARM diagrams	4767
ARM EXPRESS listing entities	4589
mapping table	2360
stratum_average_surface	
application assertion	1330

application object	1032
ARM diagrams	4757
ARM EXPRESS listing entities	4589
mapping table	2191, 2882
stratum_average_surface_shape	
application assertion	1330, 1333, 1337
application object	1032
ARM diagrams	4768
ARM EXPRESS listing entities	4589
mapping table	2361
stratum_average_surface_shape_constraint	
AIM EXPRESS expanded listing rules	4226
stratum_concept	
ARM diagrams	4629
ARM EXPRESS listing types	4430
stratum_concept_relationship	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4053
AIM EXPRESS short listing entities	3436
application assertion	1330
application object	1033
ARM diagrams	4736
ARM EXPRESS listing entities	4589
mapping table	2018, 2024, 2042, 2044
stratum_concept_to_physical_usage_view_assignment	
application assertion	1330–1331
application object	1034
ARM diagrams	4734
ARM EXPRESS listing entities	4589
mapping table	1993
stratum_feature	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4053
AIM EXPRESS short listing entities	3439
application assertion ...	1178, 1186, 1206, 1215, 1246, 1251, 1253, 1256, 1264, 1269, 1306–1307, 1311, 1317, 1326–1327, 1331, 1333–1335
application object	1035
ARM diagrams	4707
ARM EXPRESS listing entities	4590
mapping table	1925, 2082, 2091, 2093, 2192, 2904
stratum_feature_3d_shape	
application assertion	1331–1332
application object	1037
ARM diagrams	4768
ARM EXPRESS listing entities	4590
mapping table	2362

stratum_feature_3d_shape_select	
ARM diagrams	4757
ARM EXPRESS listing types	4430
stratum_feature_conductive_join	
application assertion	1166, 1332
application object	1038
ARM diagrams	4736
ARM EXPRESS listing entities	4590
mapping table	2044
stratum_feature_in_stratum_3d_position	
application assertion	1332
application object	1039
ARM diagrams	4757
ARM EXPRESS listing entities	4590
mapping table	2193, 2906
stratum_feature_in_stratum_planar_position	
application assertion	1332–1333
application object	1040
ARM diagrams	4758
ARM EXPRESS listing entities	4590
mapping table	2195, 2907
stratum_feature_non_planar_2d_shape	
application assertion	1332–1334
application object	1041
ARM diagrams	4769
ARM EXPRESS listing entities	4591
mapping table	2363
stratum_feature_non_planar_2d_shape_constraint	
AIM EXPRESS expanded listing rules	4226
stratum_feature_planar_representation_select	
ARM diagrams	4758
ARM EXPRESS listing types	4431
stratum_feature_planar_shape	
application assertion	1333, 1335
application object	1042
ARM diagrams	4769
ARM EXPRESS listing entities	4591
mapping table	2363
stratum_feature_shape_stratum_average_surface_shape_relationship	
application assertion	1333–1334
application object	1043
ARM diagrams	4769
ARM EXPRESS listing entities	4591
mapping table	2364
stratum_feature_shape_stratum_surface_shape_relationship	
application assertion	1334

application object	1044
ARM diagrams	4824
ARM EXPRESS listing entities	4591
mapping table	2364
stratum_feature_styled_area_representation	
application assertion	1332, 1334–1335
application object	1045
ARM diagrams	4770
ARM EXPRESS listing entities	4591
mapping table	2365
stratum_feature_template	
application assertion	1282, 1335, 1341
application object	1046
ARM diagrams	4799
ARM EXPRESS listing entities	4591
mapping table	2615
stratum_feature_template_component	
application assertion	1231, 1332, 1335
application object	1047
ARM diagrams	4758
ARM EXPRESS listing entities	4591
mapping table	2196
stratum_feature_unique_constraint	
AIM EXPRESS expanded listing rules	4228
stratum_interconnect_module_3d_position	
application assertion	1335–1336
application object	1048
ARM diagrams	4759
ARM EXPRESS listing entities	4592
mapping table	2198, 2914
stratum_interconnect_module_3d_position_constraint	
AIM EXPRESS expanded listing rules	4228
stratum_interconnect_module_planar_position	
application assertion	1336
application object	1049
ARM diagrams	4759
ARM EXPRESS listing entities	4592
mapping table	2201, 2917
stratum_interconnect_module_planar_position_constraint	
AIM EXPRESS expanded listing rules	4229
stratum_make_from_relationship	
application assertion	1256, 1336
application object	1050
ARM diagrams	4667
ARM EXPRESS listing entities	4592
mapping table	2204

stratum_make_from_relationship_constraint	
AIM EXPRESS expanded listing rules	4229
stratum_occurrence_specific_padstack_definition	
AIM EXPRESS short listing entities	3441
application object	1051
ARM EXPRESS listing entities	4592
mapping table	2616
stratum_planar_shape	
application assertion	1333, 1336–1337
application object	1051
ARM diagrams	4770
ARM EXPRESS listing entities	4592
mapping table	2367
stratum_planar_shape_constraint	
AIM EXPRESS expanded listing rules	4229
stratum_surface	
AIM diagrams	4890
AIM EXPRESS expanded listing entities	4054
AIM EXPRESS short listing entities	3441
application assertion	1150–1151, 1166, 1178, 1206, 1254, 1317, 1328, 1331, 1337
application object	1052
ARM diagrams	4759
ARM EXPRESS listing entities	4592
mapping table	2191, 2205, 2882
stratum_surface_designation	
ARM diagrams	4759
ARM EXPRESS listing types	4431
stratum_surface_in_stratum_3d_position	
application assertion	1337
application object	1053
ARM diagrams	4770
ARM EXPRESS listing entities	4593
mapping table	2368
stratum_surface_in_stratum_3d_position_constraint	
AIM EXPRESS expanded listing rules	4229
stratum_surface_shape	
application assertion	1334, 1337
application object	1054
ARM diagrams	4823
ARM EXPRESS listing entities	4593
mapping table	2370
stratum_surface_shape_select	
ARM diagrams	4770
ARM EXPRESS listing types	4431
stratum_surface_unique_constraint	
AIM EXPRESS expanded listing rules	4230

stratum_technology	
AIM diagrams	4863
AIM EXPRESS expanded listing entities	4057
AIM EXPRESS short listing entities	3442
application assertion	1202, 1259, 1268–1269, 1273, 1310, 1326, 1329, 1335, 1337–1338
application object	1054
ARM diagrams	4783
ARM EXPRESS listing entities	4593
mapping table	2420, 2423, 2435, 2883
stratum_technology_line_width_tolerance_assignment	
ARM diagrams	4764
stratum_technology_line_width_tolerance_assignment_constraint	
AIM EXPRESS expanded listing rules	4231
stratum_technology_link	
AIM EXPRESS short listing entities	3449
application assertion	1310, 1338
application object	1057
ARM EXPRESS listing entities	4593
mapping table	2748
stratum_type_independent_padstack_definition	
AIM EXPRESS short listing entities	3451
application object	1058
ARM EXPRESS listing entities	4594
mapping table	2617
string_property	
ARM diagrams	4660
ARM EXPRESS listing types	4431
structural_domain	11
structured_text	
application assertion	1220, 1338–1339
application object	1059
ARM diagrams	4832
ARM EXPRESS listing entities	4594
mapping table	1716
structured_text_representation_context	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4058
AIM EXPRESS short listing entities	3451
structured_text_representation_item	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	4058
AIM EXPRESS short listing entities	3452
mapping table	1716
style_context_select	
AIM diagrams	4867
AIM EXPRESS expanded listing types	3827

styled_area_representation	
ARM diagrams	4768
ARM EXPRESS listing types	4431
styled_curve_occurrence	
application assertion	1210, 1334, 1339
application object	1060
ARM diagrams	4837
ARM EXPRESS listing entities	4594
mapping table	2925
styled_item	
AIM diagrams	4931
AIM EXPRESS expanded listing entities	4058
mapping table	2428
styled_text	
application assertion	1266, 1339
application object	1061
ARM diagrams	4827
ARM EXPRESS listing entities	4594
mapping table	2926
sub_assembly_relationship	
application assertion	1339–1340
application object	1063
ARM diagrams	4627
ARM EXPRESS listing entities	4594
mapping table	1444, 1986, 2205
substitute_product	
application assertion	1339
application object	1065
ARM diagrams	4623
ARM EXPRESS listing entities	4594
mapping table	1409, 1962
substitute_product_constraint	
AIM EXPRESS expanded listing rules	4231
substrate_in_assembly	
ARM EXPRESS listing functions	4613
subtype_combination_shape_aspect	
AIM EXPRESS expanded listing rules	4236
subtype_exclusive_action_request_assignment	
AIM EXPRESS expanded listing rules	4237
subtype_exclusive_dimensional_location	
AIM EXPRESS expanded listing rules	4237
subtype_exclusive_dimensional_size	
AIM EXPRESS expanded listing rules	4237
subtype_exclusive_mapped_item	
AIM EXPRESS expanded listing rules	4237
subtype_exclusive_product_definition	

AIM EXPRESS expanded listing rules	4238
subtype_exclusive_product_definition_relationship	
AIM EXPRESS expanded listing rules	4238
subtype_exclusive_property_definition	
AIM EXPRESS expanded listing rules	4239
subtype_exclusive_property_definition_representation	
AIM EXPRESS expanded listing rules	4239
subtype_exclusive_representation	
AIM EXPRESS expanded listing rules	4239
subtype_exclusive_representation_context	
AIM EXPRESS expanded listing rules	4240
subtype_exclusive_shape_aspect_relationship	
AIM EXPRESS expanded listing rules	4240
subtype_mandatory_action	
AIM EXPRESS expanded listing rules	4240
subtype_mandatory_address	
AIM EXPRESS expanded listing rules	4241
subtype_mandatory_characterized_object	
AIM EXPRESS expanded listing rules	4241
subtype_mandatory_colour	
AIM EXPRESS expanded listing rules	4241
subtype_mandatory_composite_shape_aspect	
AIM EXPRESS expanded listing rules	4241
subtype_mandatory_date	
AIM EXPRESS expanded listing rules	4242
subtype_mandatory_externally_defined_item	
AIM EXPRESS expanded listing rules	4242
subtype_mandatory_geometric_tolerance	
AIM EXPRESS expanded listing rules	4243
subtype_mandatory_group	
AIM EXPRESS expanded listing rules	4243
subtype_mandatory_pre_defined_item	
AIM EXPRESS expanded listing rules	4243
subtype_mandatory_product_definition_usage	
AIM EXPRESS expanded listing rules	4244
subtype_mandatory_runout_zone_orientation	
AIM EXPRESS expanded listing rules	4244
subtype_mandatory_shape_representation	
AIM EXPRESS expanded listing rules	4245
subtype_mandatory_text_literal	
AIM EXPRESS expanded listing rules	4245
supplied_part_relationship	
AIM diagrams	4862
AIM EXPRESS expanded listing entities	4059
AIM EXPRESS short listing entities	3452
supplied_product_version	

application assertion	1170, 1340
application object	1066
ARM diagrams	4835
ARM EXPRESS listing entities	4595
mapping table	2874
supplied_product_version_unique_constraint	
AIM EXPRESS expanded listing rules	4246
supported_item	
AIM diagrams	4869
AIM EXPRESS expanded listing types	3827
surface	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	4059
surface_curve	
AIM diagrams	4912
AIM EXPRESS expanded listing entities	4059
surface_finish_specification	
application assertion	1340
application object	1068
ARM diagrams	4670
ARM EXPRESS listing entities	4595
mapping table	1717
surface_model	
AIM diagrams	4923
AIM EXPRESS expanded listing types	3827
surface_of_linear_extrusion	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	4059
surface_of_revolution	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	4059
surface_patch	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	4060
surface_prepped_terminal	
AIM EXPRESS short listing entities	3453
application assertion	1340
application object	1068
ARM diagrams	4815
ARM EXPRESS listing entities	4595
mapping table	2492
surface_profile_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4061
AIM EXPRESS short listing entities	3454
mapping table	1842–1843, 1911–1912

surface_replica	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	4061
surface_weights_positive	
AIM EXPRESS expanded listing functions	4288
swept_area_solid	
AIM diagrams	4925
AIM EXPRESS expanded listing entities	4061
swept_face_solid	
AIM diagrams	4935
AIM EXPRESS expanded listing entities	4062
swept_surface	
AIM diagrams	4907
AIM EXPRESS expanded listing entities	4062
symbol_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4062
symbol_representation_map	
AIM diagrams	4867
AIM EXPRESS expanded listing entities	4062
symbol_target	
AIM diagrams	4902
AIM EXPRESS expanded listing entities	4062
symmetric_shape_aspect	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4063
symmetrical_tolerance_characteristic	
application assertion	1340
application object	1069
ARM diagrams	4661
ARM EXPRESS listing entities	4595
mapping table	1653
symmetry_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4063
AIM EXPRESS short listing entities	3458
application object	1069
ARM diagrams	4691
ARM EXPRESS listing entities	4595
mapping table	1926
table_representation_item	
AIM EXPRESS short listing entities	3459
mapping table	1641, 2868
tangent	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	4063

mapping table	1926–1927
tangent_plane	
application assertion	1242
application object	1070
ARM diagrams	4692
ARM EXPRESS listing entities	4595
mapping table	1926
tangent_shape_element	
application object	1071
ARM diagrams	4692
ARM EXPRESS listing entities	4595
mapping table	1927
technology	
application assertion	1163, 1297, 1340–1341
application object	1072
ARM diagrams	4806
ARM EXPRESS listing entities	4595
mapping table	2677
template_definition	
application assertion	1274, 1287
application object	1073
ARM EXPRESS listing entities	4596
mapping table	2617
template_material_cross_section_boundary	
AIM EXPRESS short listing entities	3459
application assertion	1311, 1341
application object	1073
ARM EXPRESS listing entities	4596
mapping table	2749
template_select	
ARM EXPRESS listing types	4410
template_stratum_assignment	
AIM EXPRESS short listing entities	3460
application assertion	1282, 1341
application object	1074
ARM EXPRESS listing entities	4596
mapping table	2617
terminal_or_inter_stratum_feature	
ARM EXPRESS listing types	4410
terminal_pair_port_equivalence_relationship	
application assertion	1341–1342
application object	1075
ARM diagrams	4619
ARM EXPRESS listing entities	4596
mapping table	1397, 1677
terminal_swap_specification	

application assertion	1342
application object	1076
ARM diagrams	4678
ARM EXPRESS listing entities	4596
mapping table	1747, 1755
termination_constraint	
application assertion	1342
application object	1077
ARM diagrams	4635
ARM EXPRESS listing entities	4596
mapping table	1477
termination_or_junction	
ARM diagrams	4686
ARM EXPRESS listing types	4431
tertiary_orientation_feature	
AIM EXPRESS short listing entities	3460
application assertion	1342
application object	1078
ARM EXPRESS listing entities	4596
mapping table	2492, 2573
test_point_part_feature	
AIM EXPRESS short listing entities	3462
application object	1079
ARM EXPRESS listing entities	4597
mapping table	2493, 2573
test_requirement_allocation	
application assertion	1342–1343
application object	1079
ARM EXPRESS listing entities	4597
mapping table	1481, 2003
test_requirement_occurrence	
application assertion	1343
application object	1081
ARM EXPRESS listing entities	4597
mapping table	2821
test_select_product	
application object	1082
ARM diagrams	4620
ARM EXPRESS listing entities	4597
mapping table	1410, 1963
test_specification	
application assertion	1325
application object	1083
ARM diagrams	4633
ARM EXPRESS listing entities	4597
mapping table	1717

text	
AIM diagrams	4929
AIM EXPRESS expanded listing types	3827
ARM diagrams	4715
ARM EXPRESS listing types	4431
text_alignment	
AIM diagrams	4937
AIM EXPRESS expanded listing types	3827
text_font	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4063
text_font_family	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4063
text_font_in_family	
AIM diagrams	4932
AIM EXPRESS expanded listing entities	4063
text_literal	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4064
AIM EXPRESS short listing entities	3516
mapping table	2926
text_literal_with_associated_curves	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4064
text_literal_with_extent	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4064
text_or_character	
AIM diagrams	4867
AIM EXPRESS expanded listing types	3828
text_path	
AIM diagrams	4933
AIM EXPRESS expanded listing types	3828
text_string_representation	
AIM diagrams	4866
AIM EXPRESS expanded listing entities	4065
text_style	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4065
text_style_for_defined_font	
AIM diagrams	4933
AIM EXPRESS expanded listing entities	4065
theoretically_exact_dimension_representation	
application object	1083
ARM diagrams	4718

ARM EXPRESS listing entities	4597
mapping table	1927
thermal_component	
AIM diagrams	4888
AIM EXPRESS expanded listing entities	4065
AIM EXPRESS short listing entities	3462
application assertion	1344
application object	1084
ARM diagrams	4624
ARM EXPRESS listing entities	4597
mapping table	1428, 2207
thermal_component_shape_aspect	
AIM diagrams	4892
AIM EXPRESS expanded listing entities	4065
AIM EXPRESS short listing entities	3462
mapping table	2207
thermal_feature	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4065
AIM EXPRESS short listing entities	3463
application object	1084
ARM diagrams	4706
ARM EXPRESS listing entities	4598
mapping table	1887, 2548, 2574
thermal_isolation_component_shape_aspect	
AIM EXPRESS short listing entities	3463
mapping table	2207
thermal_isolation_laminate_component	
application assertion	1343
application object	1084
ARM EXPRESS listing entities	4598
mapping table	2207
thermal_isolation_removal_structure	
application assertion	1343
application object	1085
ARM diagrams	4836
ARM EXPRESS listing entities	4598
mapping table	2909
thermal_isolation_removal_structure_or_template	
ARM EXPRESS listing types	4431
thermal_isolation_removal_template	
application assertion	1343–1344
application object	1086
ARM EXPRESS listing entities	4598
mapping table	2619
thermal_isolation_removal_template_definition	

AIM EXPRESS short listing entities	3463
mapping table	2619
thermal_isolation_requirement_assignment	
application assertion	1343–1344
application object	1089
ARM diagrams	4764
ARM EXPRESS listing entities	4598
mapping table	2256
thermal_network	
AIM diagrams	4878
AIM EXPRESS expanded listing entities	4065
AIM EXPRESS short listing entities	3465
mapping table	1770
thermal_network_definition	
application object	1090
ARM diagrams	4681
ARM EXPRESS listing entities	4598
mapping table	1770
thermal_requirement_allocation	
application assertion	1344
application object	1090
ARM diagrams	4649
ARM EXPRESS listing entities	4598
mapping table	1571, 2257
thermal_requirement_occurrence	
application assertion	1344
application object	1091
ARM diagrams	4821
ARM EXPRESS listing entities	4598
mapping table	2821
thermodynamic_temperature_data_element	
application object	1091
ARM EXPRESS listing entities	4599
mapping table	1654
thermodynamic_temperature_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3828
thermodynamic_temperature_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4066
mapping table	1654
thermodynamic_temperature_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4066
three_d_purpose	
ARM EXPRESS listing types	4432

time_data_element	
application object	1092
ARM EXPRESS listing entities	4599
mapping table	1654
time_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3828
time_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4066
mapping table	1654
time_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4066
tolerance_characteristic	
application assertion	1344
application object	1093
ARM diagrams	4661
ARM EXPRESS listing entities	4599
mapping table	1654
tolerance_method_definition	
AIM diagrams	4898
AIM EXPRESS expanded listing types	3828
tolerance_select	
AIM diagrams	4898
AIM EXPRESS expanded listing types	3828
tolerance_specific_restraint_condition	
application assertion	1344
application object	1093
ARM diagrams	4725
ARM EXPRESS listing entities	4599
mapping table	1927
tolerance_value	
AIM diagrams	4898
AIM EXPRESS expanded listing entities	4066
AIM EXPRESS short listing entities	3517
tolerance_zone	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4067
AIM EXPRESS short listing entities	3517
application assertion	1195, 1313, 1321, 1344–1347
application object	1094
ARM diagrams	4711
ARM EXPRESS listing entities	4599
mapping table	1928
tolerance_zone_and_viewing_plane_orienting_relationship	

application assertion	1345, 1356
application object	1095
ARM diagrams	4728
ARM EXPRESS listing entities	4599
mapping table	1929
tolerance_zone_boundary	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	4067
AIM EXPRESS short listing entities	3465
application assertion	1344–1345
application object	1096
ARM diagrams	4699
ARM EXPRESS listing entities	4600
mapping table	1845, 1929–1930
tolerance_zone_boundary_definition	
application assertion	1345
application object	1097
ARM diagrams	4700
ARM EXPRESS listing entities	4600
mapping table	1929
tolerance_zone_boundary_member	
application assertion	1303, 1345–1346
application object	1098
ARM diagrams	4710
ARM EXPRESS listing entities	4600
mapping table	1929
tolerance_zone_boundary_member_offset	
application assertion	1312, 1345
application object	1099
ARM diagrams	4728
ARM EXPRESS listing entities	4600
mapping table	1930
tolerance_zone_circular_or_cylindrical_or_spherical_boundary	
application assertion	1215
application object	1100
ARM diagrams	4699
ARM EXPRESS listing entities	4600
mapping table	1930
tolerance_zone_definition	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4067
AIM EXPRESS short listing entities	3517
application assertion	1242, 1345–1346
application object	1101
ARM diagrams	4701
ARM EXPRESS listing entities	4601

mapping table	1847, 1931
tolerance_zone_definition_with_per_unit_size_specification	
application assertion	1346
application object	1103
ARM diagrams	4701
ARM EXPRESS listing entities	4601
mapping table	1931
tolerance_zone_definition_with_specified_orientation	
application assertion	1346
application object	1104
ARM diagrams	4701
ARM EXPRESS listing entities	4601
mapping table	1932
tolerance_zone_explicit_opposing_boundary_set	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	4067
AIM EXPRESS short listing entities	3467
application assertion	1312, 1346
application object	1105
ARM diagrams	4699
ARM EXPRESS listing entities	4601
mapping table	1933–1934
tolerance_zone_form	
AIM diagrams	4900
AIM EXPRESS expanded listing entities	4068
AIM EXPRESS short listing entities	3518
application assertion	1345, 1347
application object	1106
ARM diagrams	4729
ARM EXPRESS listing entities	4601
mapping table	1934
tolerance_zone_implicit_opposing_boundary_set	
AIM diagrams	4899
AIM EXPRESS expanded listing entities	4068
AIM EXPRESS short listing entities	3468
application object	1107
ARM diagrams	4699
ARM EXPRESS listing entities	4601
mapping table	1934
tolerance_zone_opposing_boundary_set	
application assertion	1275, 1312
application object	1107
ARM diagrams	4699
ARM EXPRESS listing entities	4601
mapping table	1934
tolerance_zone_per_unit_size_specification	

application assertion	1346–1347
application object	1108
ARM diagrams	4730
ARM EXPRESS listing entities	4601
mapping table	1934
toleranceable_shape_element	
ARM diagrams	4840
ARM EXPRESS listing types	4432
topological_junction	
application assertion	1255, 1294, 1347
application object	1109
ARM diagrams	4687
ARM EXPRESS listing entities	4602
mapping table	1814, 2044
topological_junction_unique_constraint	
AIM EXPRESS expanded listing rules	4247
topological_representation_item	
AIM diagrams	4919
AIM EXPRESS expanded listing entities	4068
topology_reversed	
AIM EXPRESS expanded listing functions	4289
toroidal_surface	
AIM diagrams	4908
AIM EXPRESS expanded listing entities	4068
torus	
AIM diagrams	4924
AIM EXPRESS expanded listing entities	4068
total_runout_tolerance	
AIM diagrams	4897
AIM EXPRESS expanded listing entities	4068
AIM EXPRESS short listing entities	3468
application object	1110
ARM diagrams	4691
ARM EXPRESS listing entities	4602
mapping table	1935
trace_junction_or_via	
ARM diagrams	4738
ARM EXPRESS listing types	4432
trace_template	
application assertion	1187–1188, 1347
application object	1112
ARM diagrams	4799
ARM EXPRESS listing entities	4602
mapping table	2619
transformation	
AIM diagrams	4903

AIM EXPRESS expanded listing types	3828
transition_code	
AIM diagrams	4917
AIM EXPRESS expanded listing types	3828
transmission_line_functional_unit_terminal_allocation	
application assertion	1347–1348
application object	1112
ARM diagrams	4678
ARM EXPRESS listing entities	4602
mapping table	1756
transmission_line_functional_unit_to_routed_printed_component_allocation	
application assertion	1347–1348
application object	1115
ARM diagrams	4678
ARM EXPRESS listing entities	4602
mapping table	1757
trimmed_curve	
AIM diagrams	4916
AIM EXPRESS expanded listing entities	4069
mapping table	2835, 2857
trimmed_line	
application assertion	1168, 1187, 1252, 1348
application object	1116
ARM diagrams	4824
ARM EXPRESS listing entities	4602
mapping table	2857
trimmed_or_bounding_curve	
ARM diagrams	4746
ARM EXPRESS listing types	4432
trimming_preference	
AIM diagrams	4916
AIM EXPRESS expanded listing types	3829
trimming_select	
AIM diagrams	4916
AIM EXPRESS expanded listing types	3829
true_profile	
application assertion	1312, 1348–1349
application object	1116
ARM diagrams	4729
ARM EXPRESS listing entities	4603
mapping table	1935
true_profile_representations	
ARM diagrams	4729
ARM EXPRESS listing types	4432
two_direction_repeat_factor	
AIM diagrams	4931

AIM EXPRESS expanded listing entities	4069
two_or_three_dimensional	
ARM diagrams	4825
ARM EXPRESS listing types	4432
type_qualifier	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	4069
application object	1117
ARM EXPRESS listing entities	4603
mapping table	1655, 1908, 1937–1938
uncertainty_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4069
uncertainty_qualifier	
AIM diagrams	4889
AIM EXPRESS expanded listing entities	4069
application object	1118
ARM EXPRESS listing entities	4603
mapping table	1655
uniform_curve	
AIM diagrams	4915
AIM EXPRESS expanded listing entities	4069
uniform_surface	
AIM diagrams	4910
AIM EXPRESS expanded listing entities	4069
unique_characterized_product_category	
AIM EXPRESS short listing rules	3556
unique_version_change_order	
AIM EXPRESS expanded listing functions	4289
unit	
AIM diagrams	4894
AIM EXPRESS expanded listing types	3829
unrouted_conductive_interconnect_element	
application assertion	1349
application object	1118
ARM EXPRESS listing entities	4603
mapping table	2208
unrouted_join_relationship	
application object	1119
ARM EXPRESS listing entities	4603
mapping table	2045
unsupported_passage	
application assertion	1349
application object	1119
ARM diagrams	4760
ARM EXPRESS listing entities	4603

mapping table	2208
unsupported_passage_dependent_non_functional_land	
application assertion	1349
application object	1120
ARM diagrams	4775
ARM EXPRESS listing entities	4603
mapping table	2394
unsupported_passage_template	
application assertion	1193, 1349
application object	1121
ARM diagrams	4732
ARM EXPRESS listing entities	4604
mapping table	1948, 2441
usage_concept	
ARM diagrams	4800
ARM EXPRESS listing types	4433
usage_concept_usage_relationship	
application assertion	1349–1351
application object	1121
ARM diagrams	4800
ARM EXPRESS listing entities	4604
mapping table	2574, 2619, 2679, 2750
usage_view_connection_zone_terminal_shape_relationship	
AIM EXPRESS short listing entities	3469
application assertion	1191, 1352–1354
application object	1123
ARM EXPRESS listing entities	4604
mapping table	2537, 2583
usage_view_terminal	
ARM EXPRESS listing types	4433
user_defined_dimension_qualifier	
application object	1124
ARM diagrams	4715
ARM EXPRESS listing entities	4604
mapping table	1937
user_defined_geometric_tolerance_qualifier	
application object	1125
ARM diagrams	4721
ARM EXPRESS listing entities	4604
mapping table	1938
using_items	
AIM EXPRESS expanded listing functions	4290
using_representations	
AIM EXPRESS expanded listing functions	4291
valid_2d_wireframe_edge_curve	
AIM EXPRESS expanded listing functions	4291

valid_basis_curve_in_2d_wireframe	
AIM EXPRESS expanded listing functions	4293
valid_calendar_date	
AIM EXPRESS expanded listing functions	4293
valid_constraining_shape	
ARM EXPRESS listing functions	4614
valid_geometrically_bounded_wf_curve	
AIM EXPRESS expanded listing functions	4294
valid_geometrically_bounded_wf_point	
AIM EXPRESS expanded listing functions	4294
valid_measure_value	
AIM EXPRESS expanded listing functions	4295
valid_time	
AIM EXPRESS expanded listing functions	4295
valid_units	
AIM EXPRESS expanded listing functions	4297
valid_wireframe_edge_curve	
AIM EXPRESS expanded listing functions	4298
valid_wireframe_vertex_point	
AIM EXPRESS expanded listing functions	4299
value_qualifier	
AIM diagrams	4889
AIM EXPRESS expanded listing types	3829
ARM EXPRESS listing types	4433
vector	
AIM diagrams	4906
AIM EXPRESS expanded listing entities	4070
application assertion	1234–1235
application object	1126
ARM diagrams	4832
ARM EXPRESS listing entities	4605
mapping table	2857
vector_difference	
AIM EXPRESS expanded listing functions	4300
vector_or_direction	
AIM diagrams	4906
AIM EXPRESS expanded listing types	3829
versioned_action_request	
AIM diagrams	4869
AIM EXPRESS expanded listing entities	4070
AIM EXPRESS short listing entities	3518
mapping table	1679–1680, 1685
versioned_action_request_requires_approval	
AIM EXPRESS expanded listing rules	4249
versioned_action_request_requires_date_or_date_and_time	
AIM EXPRESS expanded listing rules	4249

versioned_action_request_requires_person_organization	
AIM EXPRESS expanded listing rules	4249
versioned_action_request_requires_status	
AIM EXPRESS expanded listing rules	4249
vertex	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	4070
vertex_loop	
AIM diagrams	4921
AIM EXPRESS expanded listing entities	4070
vertex_point	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	4070
vertex_shell	
AIM diagrams	4923
AIM EXPRESS expanded listing entities	4070
vertical_justification	
ARM diagrams	4752
ARM EXPRESS listing types	4434
via	
application assertion	1245, 1251, 1262, 1354–1356
application object	1126
ARM diagrams	4739
ARM EXPRESS listing entities	4605
mapping table	2209
via_and_contact_size_dependent_land	
application assertion	1354
application object	1128
ARM diagrams	4775
ARM EXPRESS listing entities	4605
mapping table	2396
via_and_contact_size_dependent_non_functional_land	
application assertion	1354–1355
application object	1129
ARM diagrams	4775
ARM EXPRESS listing entities	4605
mapping table	2398
via_dependent_land	
application assertion	1355
application object	1130
ARM diagrams	4775
ARM EXPRESS listing entities	4605
mapping table	2400
via_dependent_non_functional_land	
application assertion	1355
application object	1131

ARM diagrams	4777
ARM EXPRESS listing entities	4605
mapping table	2401
via_template	
application assertion	1354–1356
application object	1132
ARM diagrams	4731
ARM EXPRESS listing entities	4605
mapping table	1948, 2442
via_template_terminal	
application assertion	1176, 1350, 1353, 1355–1356
application object	1133
ARM diagrams	4733
ARM EXPRESS listing entities	4606
mapping table	1949
via_template_terminal_unique_constraint	
AIM EXPRESS expanded listing rules	4250
via_terminal	
application assertion	1231, 1260, 1356
application object	1135
ARM diagrams	4760
ARM EXPRESS listing entities	4606
mapping table	2209
view_level_non_feature_shape_definition	
application assertion	1356–1357
application object	1136
ARM EXPRESS listing entities	4606
mapping table	2857
view_shape_select	
ARM EXPRESS listing types	4434
viewing_plane	
AIM diagrams	4891
AIM EXPRESS expanded listing entities	4070
AIM EXPRESS short listing entities	3470
application assertion	1345, 1356
application object	1137
ARM diagrams	4711
ARM EXPRESS listing entities	4606
mapping table	1938
viewing_plane_based_orientation	
application assertion	1346, 1356
application object	1137
ARM diagrams	4730
ARM EXPRESS listing entities	4606
mapping table	1938
viewing_plane_definition	

application assertion	1356
application object	1138
ARM diagrams	4730
ARM EXPRESS listing entities	4606
mapping table	1939
volume_data_element	
application object	1139
ARM EXPRESS listing entities	4607
mapping table	1655
volume_measure	
AIM diagrams	4896
AIM EXPRESS expanded listing types	3829
volume_measure_with_unit	
AIM diagrams	4895
AIM EXPRESS expanded listing entities	4071
mapping table	1655
volume_shape_intersection	
AIM EXPRESS short listing entities	3471
application assertion	1198, 1357–1358
application object	1140
ARM EXPRESS listing entities	4607
mapping table	2861
volume_shape_plane_projection	
application assertion	1358
application object	1141
ARM EXPRESS listing entities	4607
mapping table	2863
volume_shape_point_projection	
application assertion	1358–1359
application object	1142
ARM EXPRESS listing entities	4607
mapping table	2865
volume_shape_projection	
ARM diagrams	4833
volume_unit	
AIM diagrams	4894
AIM EXPRESS expanded listing entities	4071
wire_shell	
AIM diagrams	4920
AIM EXPRESS expanded listing entities	4071
wire_terminal	
AIM EXPRESS short listing entities	3473
application assertion	1359
application object	1143
ARM diagrams	4822
ARM EXPRESS listing entities	4607

mapping table	2493
wireframe_model	
AIM diagrams	4918
AIM EXPRESS expanded listing types	3829
work_item	
AIM diagrams	4871
AIM EXPRESS expanded listing types	3829
AIM EXPRESS short listing types	2962
work_order	
application assertion	1359–1360
application object	1144
ARM diagrams	4664
ARM EXPRESS listing entities	4608
mapping table	1680
work_order_unique_constraint	
AIM EXPRESS expanded listing rules	4251
work_request	
application assertion	1360–1362
application object	1147
ARM diagrams	4665
ARM EXPRESS listing entities	4608
mapping table	1685
work_request_unique_constraint	
AIM EXPRESS expanded listing rules	4251
year_number	
AIM diagrams	4883
AIM EXPRESS expanded listing types	3829