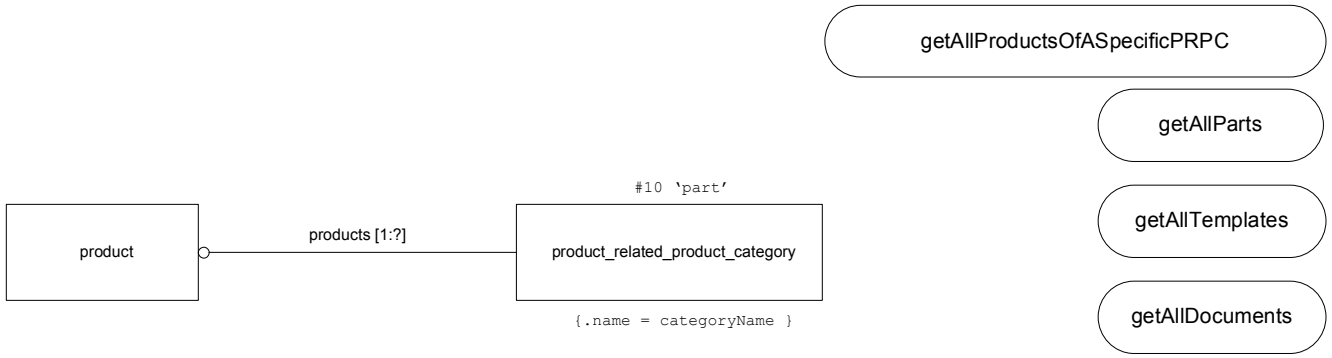
getAllProductsOfASpecificPRPC	*Returns an aggregate of all product entities in the model that are associated with a given product_related_product_category.*
isProductInASpecificPRPC	*Returns true if there exists a product_related_product_category of the specified categoryName referencing the given product through its products attribute*
getAllProductsWithGivenAssignedClass	*Returns an aggregate of all product entities in the model that are associated with a given class.*
getAllAssigningClassForProduct	*Returns an aggregate of all product entities in the model that are assigned to a specified class*
getAllParameterAssignmentsForPart	*Returns an aggregate of all parameter_assignment entities that are associated with a given product.*
getAllProductDefinitionsOfSpecifiedProductDefinitionContextRole	*Returns an aggregate of product definitions that have a given specified product_definition_context_role and frame of reference related through a product_definition_context_association*
getQualifiedProductDefinitions	*Returns an aggregate of all product_definition entities in the model that are instances of products with a given specified product_definition_context_role and frame of reference related through a product_definition_context_association.*
getUsageViewOfProductDefinition	*Returns the product_definition corresponding to the usage view of the given product_definition. For example, returns the pcb usage view given the pcb (design view) or the pca usage view given the pca (design view).*
getPcas	*Return all entities that satisfy the mapping requirements for the design view of a pca*
getPcbs	*Return all entities that satisfy the mapping requirements for the design view of a pcb*
getPanels	*Return all entities that satisfy the mapping requirements for the design view of a panel.*
getInterconnectModuleComponents	*Returns an aggregate of all interconnect_module_component entities in the model that are instances of products with associated characterized_class of 'interconnect'*
getAllInterconnectModuleComponentsInPanel	*Returns an aggregate of all Interconnect_module_components (instances of a pcb) that are located in an* layered_interconnect_panel_design_view (panel design view).
getAllInterconnectModuleComponentsInPCA	Returns an aggregate of all interconnect_module_components (instances of a pcb) that are located in a layered_assembly_module_design_view (PCA design view).
getAllPackagedComponentsInAssembly	*Returns an aggregate of all packaged_components within a given PCA.*
getProductOfPackagedComponent	*Returns the product of which the given packaged_component is an instance*
getPackageOfPackagedComponent	*Returns the package that is used by a particular packaged_component.*
getShapeRepresentationOfPackageWithSpecifiedPurpose	*Returns the shape_representation that of the given package with a specified 'predefined shape_purpose.'*
getShapeRepresentationWithSpecifiedPurpose	Returns the shape_representation within the given set of shape_representations that has a specified 'predefined shape purpose
getAllKeepoutsForPhysicalUnitShapeModel	Given a shape_representation corresponding to a mapping of ARM AO Physical_unit_planar_shape_model, this method will return all associated shape_representations corresponding to mappings of Physical_unit_planar_keepout_shape_model
getKeepoutShapeRepresentationWithSpecifiedKeepoutCategory	Given an aggregate of shape_representation corresponding to Physical_unit_planar_keepout_shape_model, returns the first shape_representation whose associated keepout_design_object_category matches the given description
getAllAssemblyJointsInPca	*Returns an aggregate of all assembly_joint in an assembly_definition. There is an assembly joint for each terminal of a packaged_part.*
getJoinTerminalForAssemblyJoint	*Returns the packaged_part_terminal ('join terminal') for a given assembly_joint.*
getLaminateComponentForAssemblyJoint	Returns the associated laminate_component for a given assembly_joint.
getLaminateComponentForIMCT	Returns the associated laminate_component for a given interconnct_module_component_terminal
getIMITforAC	Returns the associated imterconnect_module_interface_terminal for a given assembly_component if applicable
getPartToolingFeaturesInPcb	*Returns an aggregate of part_tooling_features that are located on the interconnect_definition (Pcb). This includes part_tooling_features and its subtype fiducial_part_feature.*
getLaminateComponentForPartToolingFeature	*Returns an associated laminate_component for a part_tooling_feature (or its subtype fiducial_part_feature).*



// Returns an aggregate of all product entities in the model that are associated with a given product_related_product_category

```
Aggregate<product> getAllProductsOfASpecificPRPC(Model m, String categoryName)
{
    Aggregate<product> a_productsOfCategory = null

    Aggregate<product_related_product_category> a_prpc = allInstancesOp(m)
        where {product_related_product_category prpc}
              {prpc.name = categoryName}

    For Each product_related_product_category prpc Of a_prpc
    {
        Aggregate<product> a_product = referencedEntititesOp(prpc)
            where {product p}
                  {prpc.products Contains p}

        For Each product p Of a_product
        {
            Add p to a_productsOfCategory
        }
    }
    return a_productsOfCategory
}
```

// Returns an aggregate of all product entities in the model satisfying the MIM mapping of the ARM AO Template.
// These products have an associated product_related_product_category of 'template model'

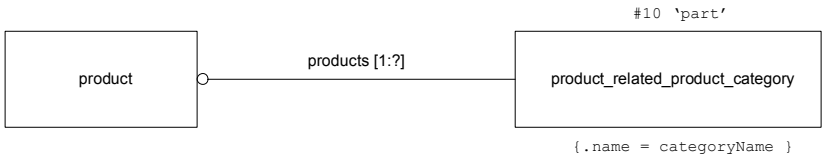
```
Aggregate<product> getAllTemplates(Model m)
{
    return getAllProductsOfASpecificPRPC(Model m, 'template model')
}
```

// Returns an aggregate of all product entities in the model satisfying the MIM mapping of the ARM AO Document.
// These products have an associated product_related_product_category of 'document'.

```
Aggregate<product> getAllDocuments(Model m)
{
    return getAllProductsOfASpecificPRPC(Model m, 'document')
}
```

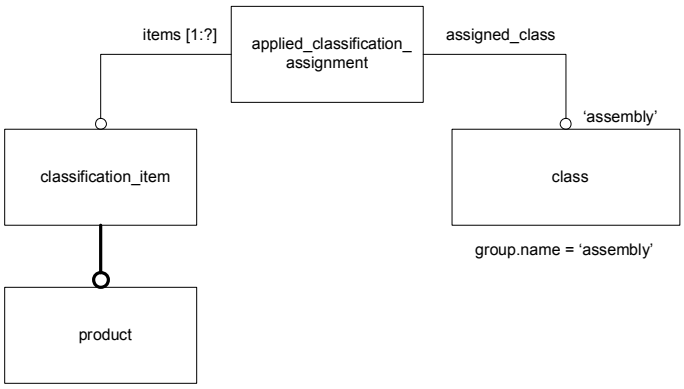
// Returns an aggregate of all product entities in the model satisfying the MIM mapping of the ARM AO Part.
// Note: raw materials are not included in the results of this query.
// These products have an associated product_related_product_category of 'part'.

```
Aggregate<product> getAllParts(Model m)
{
    return getAllProductsOfASpecificPRPC(Model m, 'part')
}
```



// Returns true if there exists a product_related_product_category of the specified categoryName referencing the given product
// through its products attribute.

```
boolean isProductOfASpecificPRPC(product p, String categoryName)
{
    Aggregate<product> a_productsOfCategory = null

    Aggregate<product_related_product_category> a_prpc = referencingEntitiesOp(p)
        where {product_related_product_category prpc}
              {prpc.products Contains p}

    For Each product_related_product_category prpc Of a_prpc
    {
        If (prpc.name = categoryName)
            return true
    }
    return false
}
```



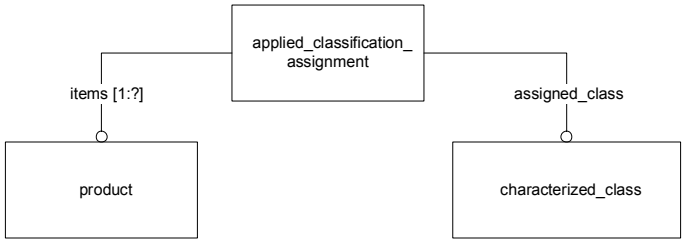
// Returns an aggregate of all product entities in the model that are assigned to a specified class

Aggregate<product> getAllProductsWithGivenAssignedClass(Model m, String className)
{
    Aggregate<product> a_productsOfCategory = null

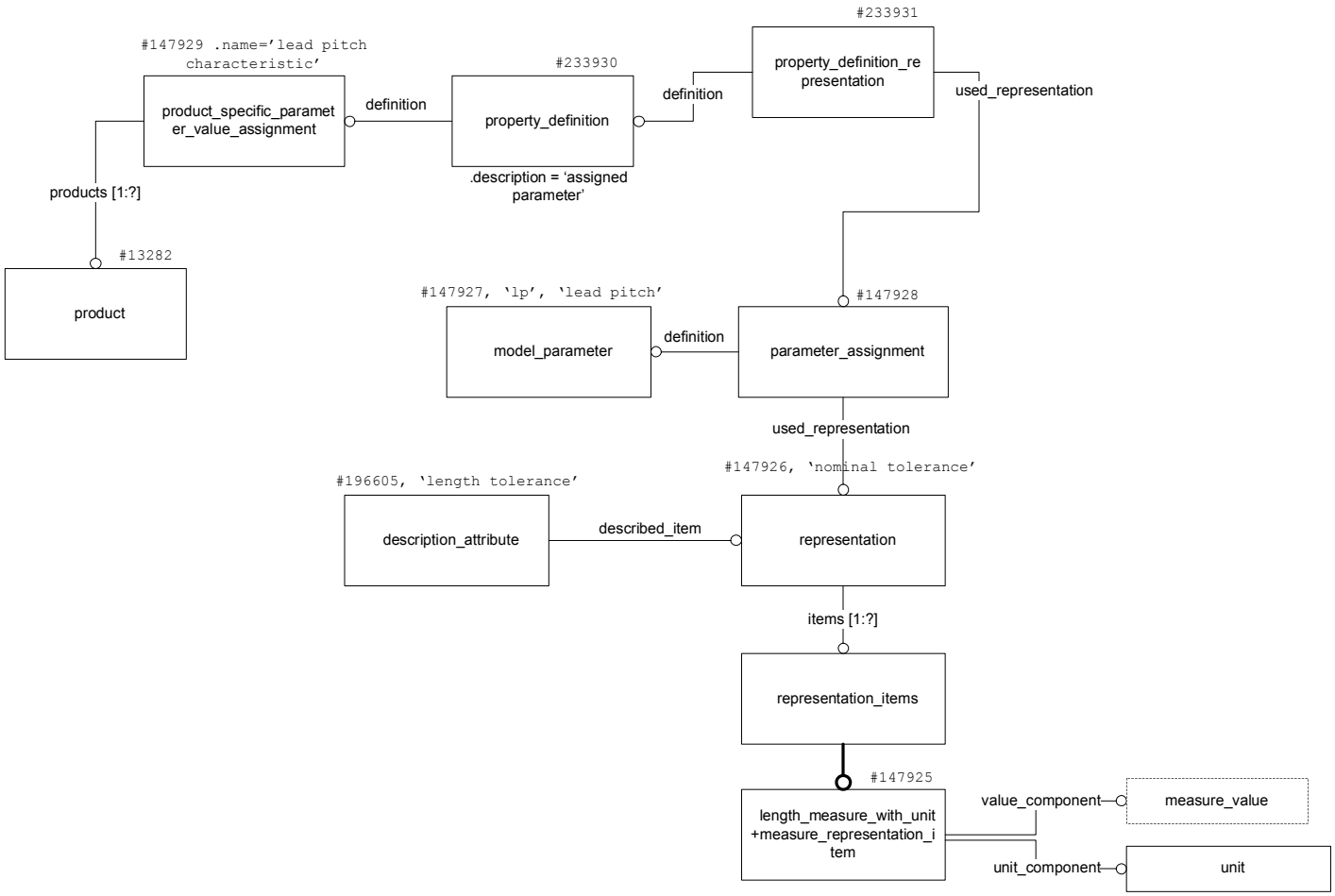
    Aggregate<product> a_p = allInstancesOp(m)
        where {product p}
              {p Contained in m}

    For Each product p Of a_p
    {
        Aggregate<class> a_c = relatedEntitiesOp(p)
            where {applied_classification_assignment aca}
                  {class e_c}
                  {aca.items Contains p}
                  {aca.assigned_class->e_c}

        For Each class e_c Of a_c
        {
            If (e_c.name = className)
            {
                Add p to a_productsOfClass
            }
        }
    }
    return a_productsOfClass
}




```
// Returns an aggregate of all instances of class that are associated with a given product through an
// applied_classification_assignement

Aggregate<class > getAllAssigningClassForProduct(product p)
{
    Aggregate<class > a_c = relatedEntititesOp(p)
        where  {applied_classification_assignment aca}
               {class e_c}
               {aca.items Contains p}
               {aca.assigned_class->e_c}

    return a_c
}
```



// Returns an aggregate of all parameter_assignment entities that are associated with a given product

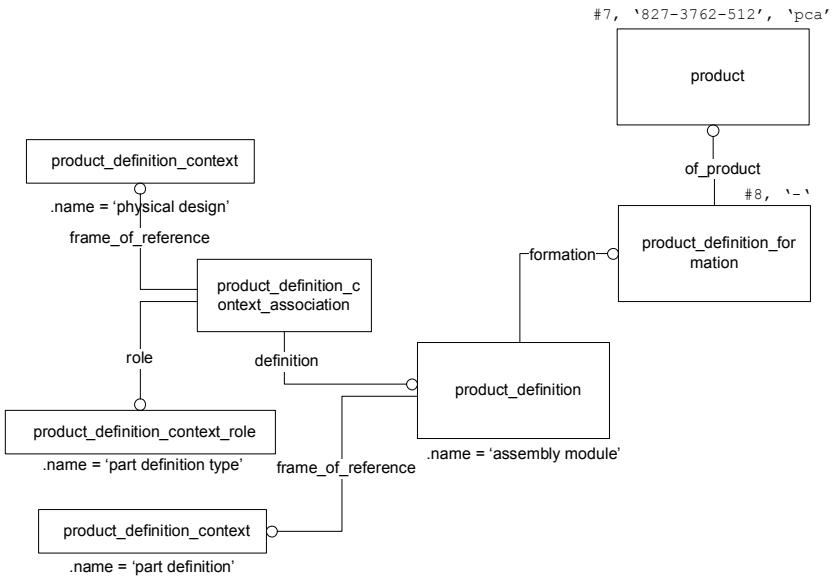
Aggregate<parameter_assignment > getAllParameterAssignmentsForProduct(product p)
{
    Aggregate<parameter_assignment> a_pa = null

    Aggregate<product_specific_parameter_value_assignment> a_pspva = referencingEntitiesOp(p)
        where {product_specific_parameter_value_assignment pspva}
            {pspva.products Contains p}

    For Each product_specific_parameter_value_assignment pspva Of a_pspva
    {
        property_definition pd = referencingEntityOp(pspva)
            where {pd.definition->pspva}
                {pd.description = 'assigned parameter'}

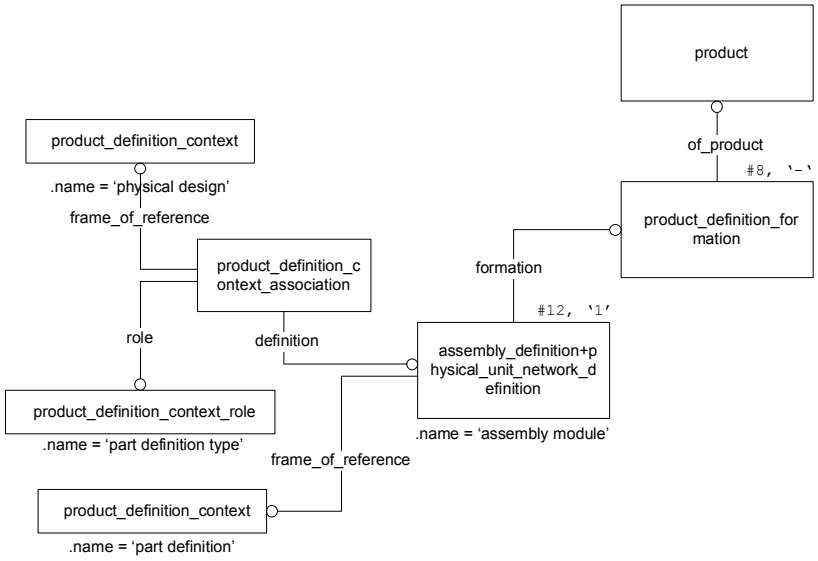
        p*arameter_assignment* pa = relatedEntityOp(pd)
            where {property_definition_representation pdr}
                {pd<-pdr.definition}
                {pdr.used_representation->pa}

        Add pa to a_pa
    }
    return a_pa
}



// Given an aggregate of product_definition, returns a subset of the given aggregate that are qualified by a
// specified product_definition_context_role and product_definition_context (frame of reference)
// The *product_definition_context_role and product_definition_context*
// are related to the product_definition through a product_definition_context_association

```
Aggregate<product_definition> getAllProductDefinitionsOfSpecifiedProductDefinitionContextRole(
        Aggregate<product_definition> a_pd, String contextRole, String contextRoleFrameOfReference)
{
    Aggregate<product_definition> a_qualifyingProductDefinition = null

    For Each product_definition e_pd of a_pd
    {
        Aggregate<product_definition_context_association> a_pdca = referencingEntitiesOp(e_pd)
            where {product_definition_context_association e_pdca}
                  {e_pdca.definition->e_pd}

        For Each product_definition_context_association e_pdca of a_pdca
        {
            product_definition_context_role e_pdcr = referencedEntityOp(e_pdca)
                where {e_pdca.role -> e_pdcr}
                      {e_pdcr.name = contextRole}

            If e_pdcr != null
            {
                product_definition_context e_pdc = referencedEntityOp(e_pdca)
                where {e_pdca.frame_of_reference -> e_pdc}
                      {e_pdc.name = contextRoleFrameOfReference}

                if e_pdc != null
                    Add e_pd to a_qualifyingProductDefinition
            }
        }
    }
    return a_qualifyingProductDefinition
}
```



// Returns an aggregate of all product_definition entities in the model that are instances of products with
// a given specified product_definition_context_role and frame
// of reference related through a product_definition_context_association.

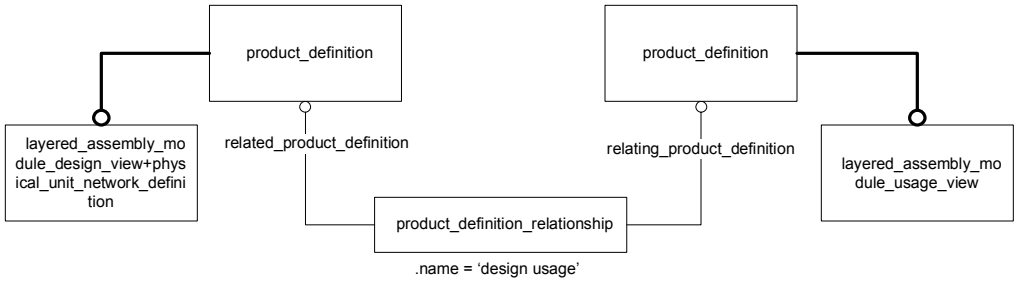
Aggregate<product_definition> getQualifiedProductDefinitions(
String contextRole,
String contextRoleFrameOfReference)
{
Aggregate<product_definition> a_pd = new *Aggregate*
*Aggregate<p*roduct> a_products = all product in model

For each product e_p in a_products
{
product_definition_formation e_pdf = referencingEntityOp(e_p)
where {e_pdf.of_product->e_p}

Aggregate<product_definition> a_pd1 = referencingEntititesOp(e_pdf)
where {product_definition pd}
{pd.formation -> e_pdf}

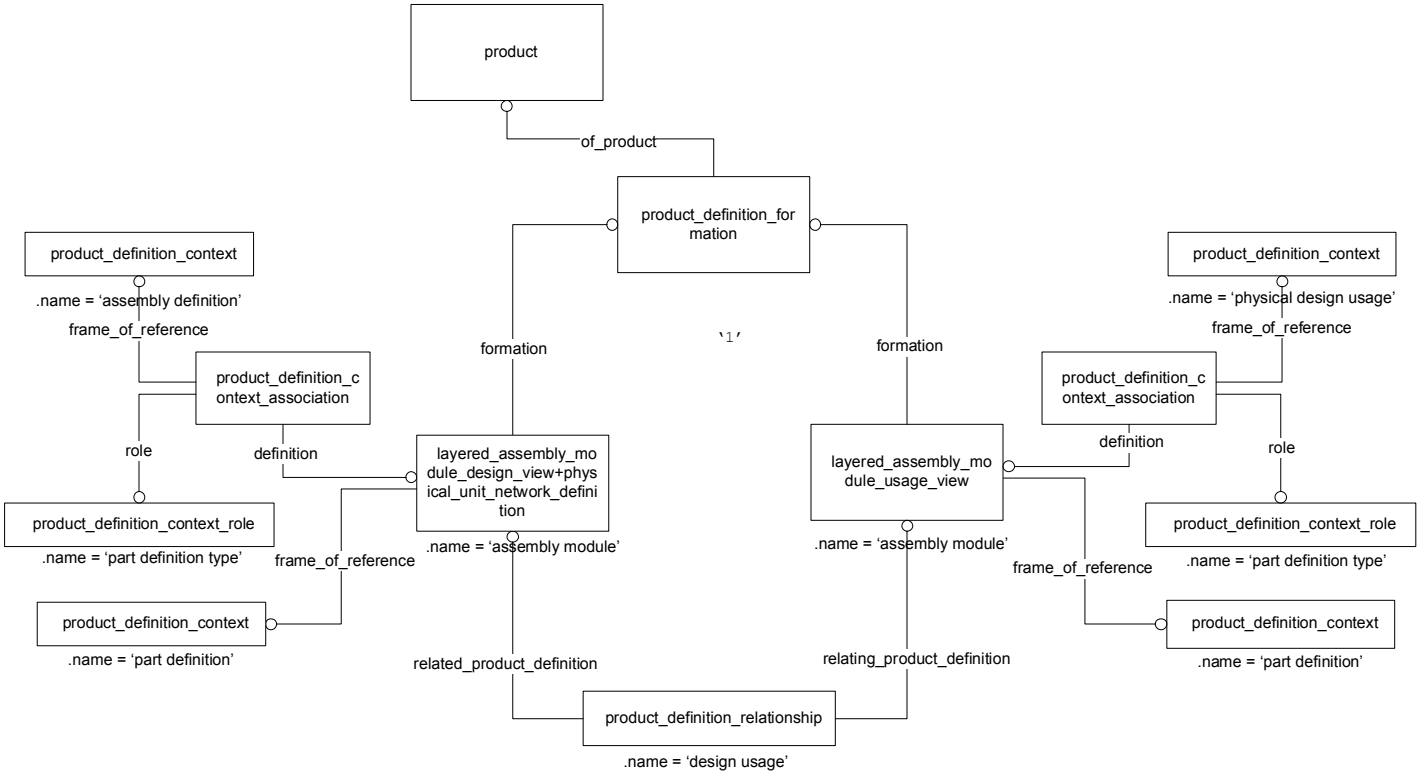
Aggregate<product_definition> a_pd2 = getAllProductDefinitionsOfSpecifiedProductDefinitionContextRole(
a_pd1, contextRole, contextRoleFrameOfReference)

Add all members of e_pd to a_pd
}
return a_pd
}



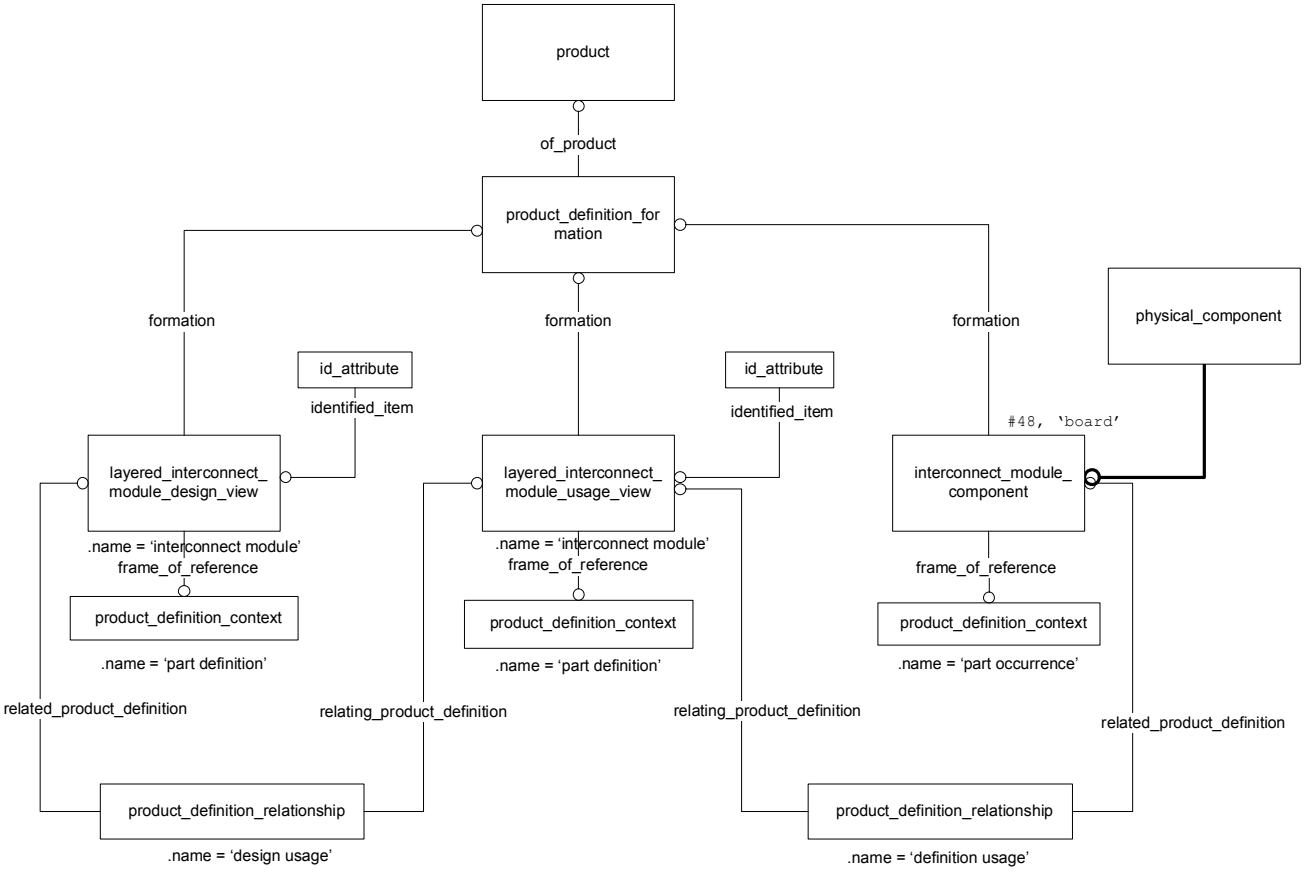
// Returns the product_definition corresponding to the usage view of the given product_definition.
// For example, returns the pcb usage view given the pcb (design view) or the pca usage view given the
// pca (design view).

product_definition getUsageViewOfProductDefinition(product_definition e_pd)
{
    product_definition e_usageView = relatedEntityOp(e_pd)
        where {product_definition_relationship pdr}
            {pdr.name = 'design usage'}
            {pdr.related_product_definition->e_pd}
            {pdr.relating_product_definition->e_usageView}
    return e_usageView;
}



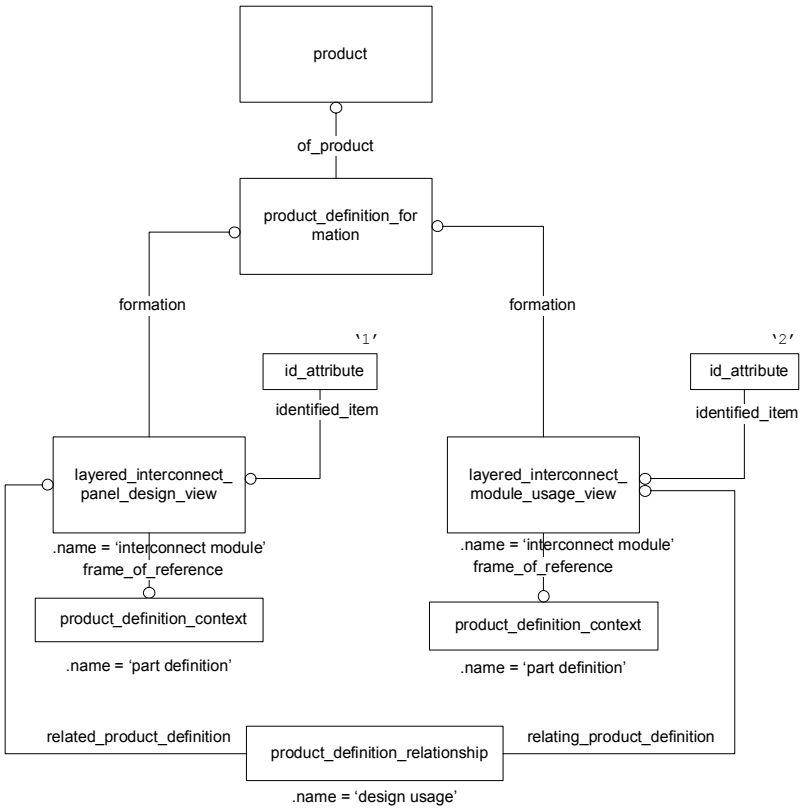
// Returns an aggregate of all layered_assembly_module_design_view entities in the model that
// have a given specified product_definition_context_role of
// 'part definition type' and frame of reference 'physical design' related through a product_definition_context_association.
// These entities represent the design view of a pca.

```
Aggregate<assembly_definition> getPcas()
{
    Aggregate<product_definition> a_pd = getQualifiedProductDefinitions(
            "part definition type", "physical design")

    Aggregate<assembly_definition> a_pca = new Aggregate<assembly_definition>

    For each product_definition e_pd in a_pd
    {
        If (e_pd is instance of layered_assembly_module_design_view)
            Add e_pd to a_pca
    }
    return a_pca
}
```

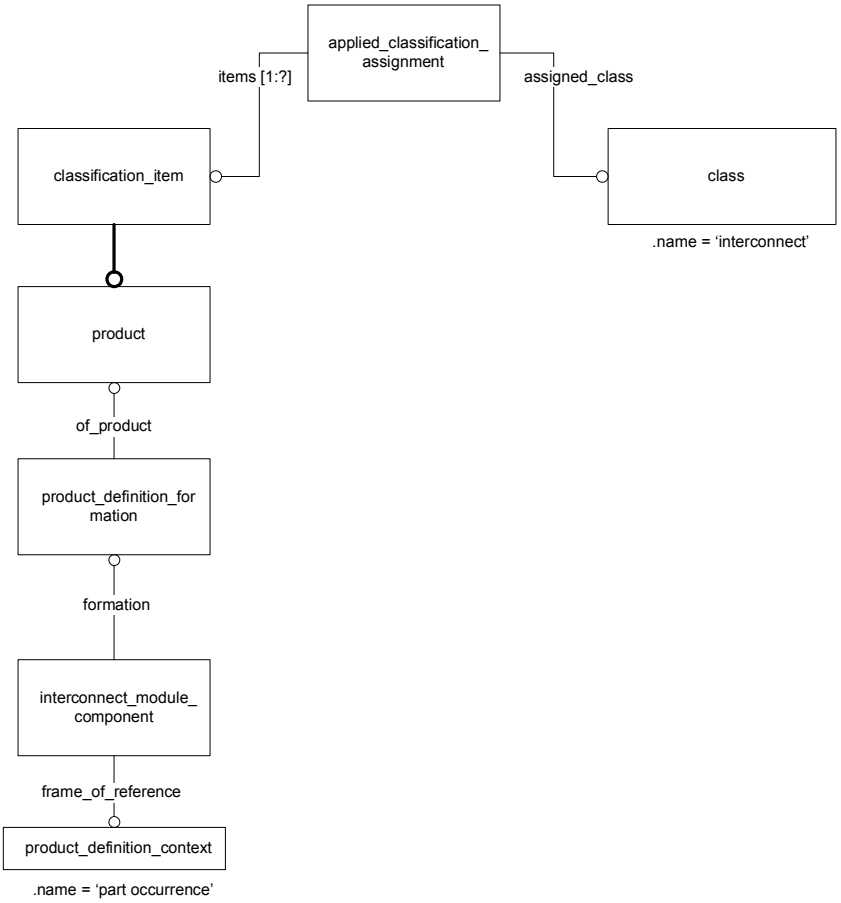


// Returns an aggregate of all layered_interconnect_module_design_view (previously interconnect_definition) entities in the model that
// have a given specified product_definition_context_role of
// 'part definition type' and frame of reference 'physical design' related through a product_definition_context_association.
// These entitites represent the design view of a pcb.
// Note: panel design view will be a layered_interconnect_panel_design_view not a layered_interconnect_module_design_view

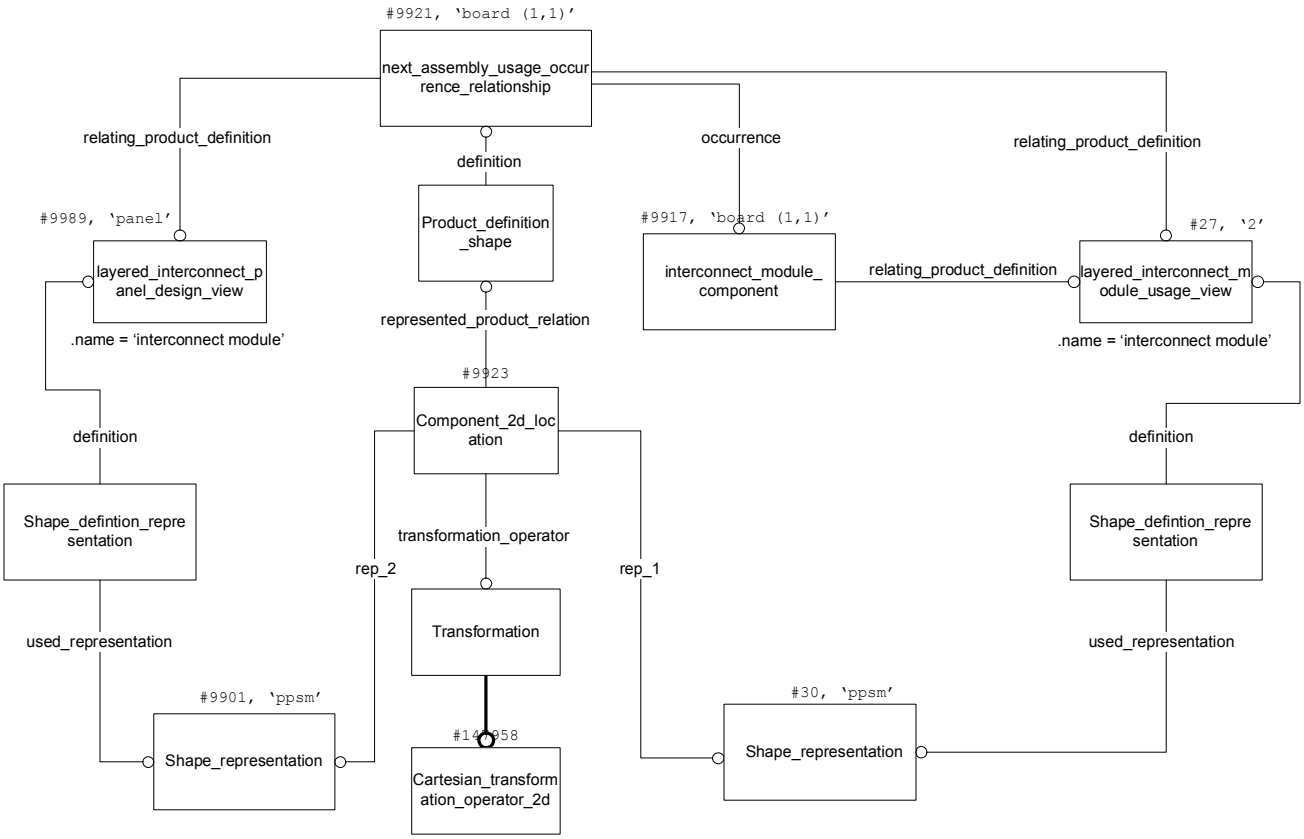
```
Aggregate<interconnect_definition > getPcbs()
{
    Aggregate<product_definition> a_pd = getQualifiedProductDefinitions(
            "part definition type", "physical design");

    Aggregate<layered_interconnect_module_design_view > a_pcb = new Aggregate<layered_interconnect_module_design_view >

    For each product_definition e_pd in a_pd
    {
        if (e_pd is instance of layered_interconnect_module_design_view )
        {
            Add e_pd to a_pcb
        }
    }
    return a_pcb;
}
```



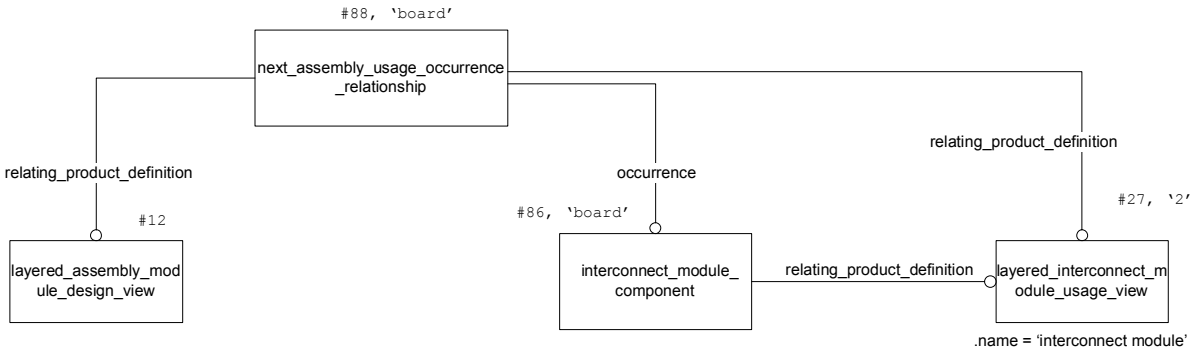

```
// Returns an aggregate of all layered_interconnect_panel_design_view entities in the model that
// have a given specified product_definition_context_role of
// 'part definition type' and frame of reference 'physical design' related through a product_definition_context_association.
// These entitites represent the design view of a pcb.
// Note: panels are included based on id string containing 'panel' which is a convention-based work-around.

Aggregate<layered_interconnect_panel_design_view> getPcbs()
{
    Aggregate<product_definition> a_pd = getQualifiedProductDefinitions(
            "part definition type", "physical design");

    Aggregate<layered_interconnect_panel_design_view> a_panel = new Aggregate<layered_interconnect_panel_design_view>

    For each product_definition e_pd in a_pd
    {
        if (e_pd is instance of layered_interconnect_panel_design_view)
        {
            Add e_pd to a_panel
        }
    }
    return a_panel;
}
```



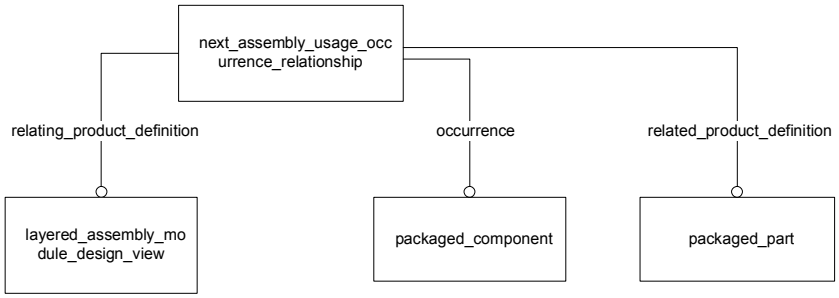

```
// Returns an aggregate of all interconnect_module_component entities in the model that are instances of products with
// associated class of 'interconnect'

Aggregate<interconnect_module_component > getInterconnectModuleComponents(Model m)
{
    Aggregate<interconnect_module_component > a_imc = null

    Aggregate<product> a_productsOfCategory = getAllProductsWithGivenAssignedClass('interconnect')

    For Each product p Of a_productsOfCategory
    {
        product_definition_formation pdf = referencingEntityOp(p)
            where {pdf.of_product->p}

        Aggregate<interconnect_module_component> a_imc2 = referencingEntitiesOp(pdf)
            where  {interconnect_module_component imc}
                   {imc.formation->pdf}

         Add All Elements of a_imc2 to a_imc
    }
    return a_imc
}
```




```
// Returns an aggregate of all interconnect_module_components (instances of a pcb) that
// are located in a layered_interconnect_panel_design_view (panel design view).
// The interconnect_module_components are instances of a product. To locate these IMCs in
// the panel, the associated shape_representation of the pcb usage view must be obtained.

Aggregate<Interconnect_module_component> getAllInterconnectModuleComponentsInPanel(
                                                    layered_interconnect_panel_design_view e_panel)
{
    Aggregate<Interconnect_module_component> a_imc = relatedEntitiesOp(e_panel)
        where {interconnect_module_component e_imc}
              {next_assembly_usage_occurrence_relationship e_nauor}
              {e_panel <- e_nauor.relating_product_definition}
              {e_nauor.occurrence -> e_imc}

    return a_imc
}
```



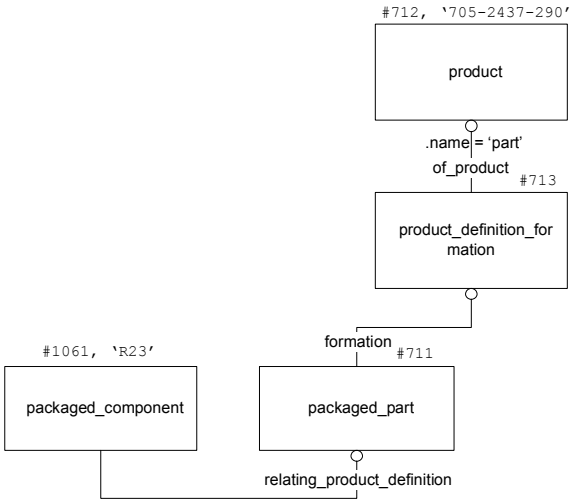

```
// Returns an aggregate of all interconnect_module_components (instances of a pcb) that
// are located in a layered_assembly_module_design_view (PCA design view).
// The interconnect_module_components are instances of a product. To locate these IMCs in
// the PCA, the associated shape_representation of the pcb usage view must be obtained.

Aggregate<Interconnect_module_component> getAllInterconnectModuleComponentsInPCA(
                                                    layered_assembly_module_design_view e_pca)
{
    Aggregate<Interconnect_module_component> a_imc = relatedEntitiesOp(e_pca)
        where  {interconnect_module_component e_imc}
               {next_assembly_usage_occurrence_relationship e_nauor}
               {e_pca <- e_nauor.relating_product_definition}
               {e_nauor.occurrence -> e_imc}

    return a_imc
}
```



// Returns an aggregate of all packaged_component that are occurrences in an layered_assembly_module_design_view (pca).

Aggregate<packaged_component> getAllPackagedComponentsInAssembly(layered_assembly_module_design_view ad)
{
    Aggregate<packaged_component> a_pc = relatedEntitiesOp(ad)
        where {next_assembly_usage_occurrence_relationship nauou}
                {packaged_component pc}
                {naour.relating_product_definition->ad}
                {naour.occurrence->pc}

    return a_pc
}



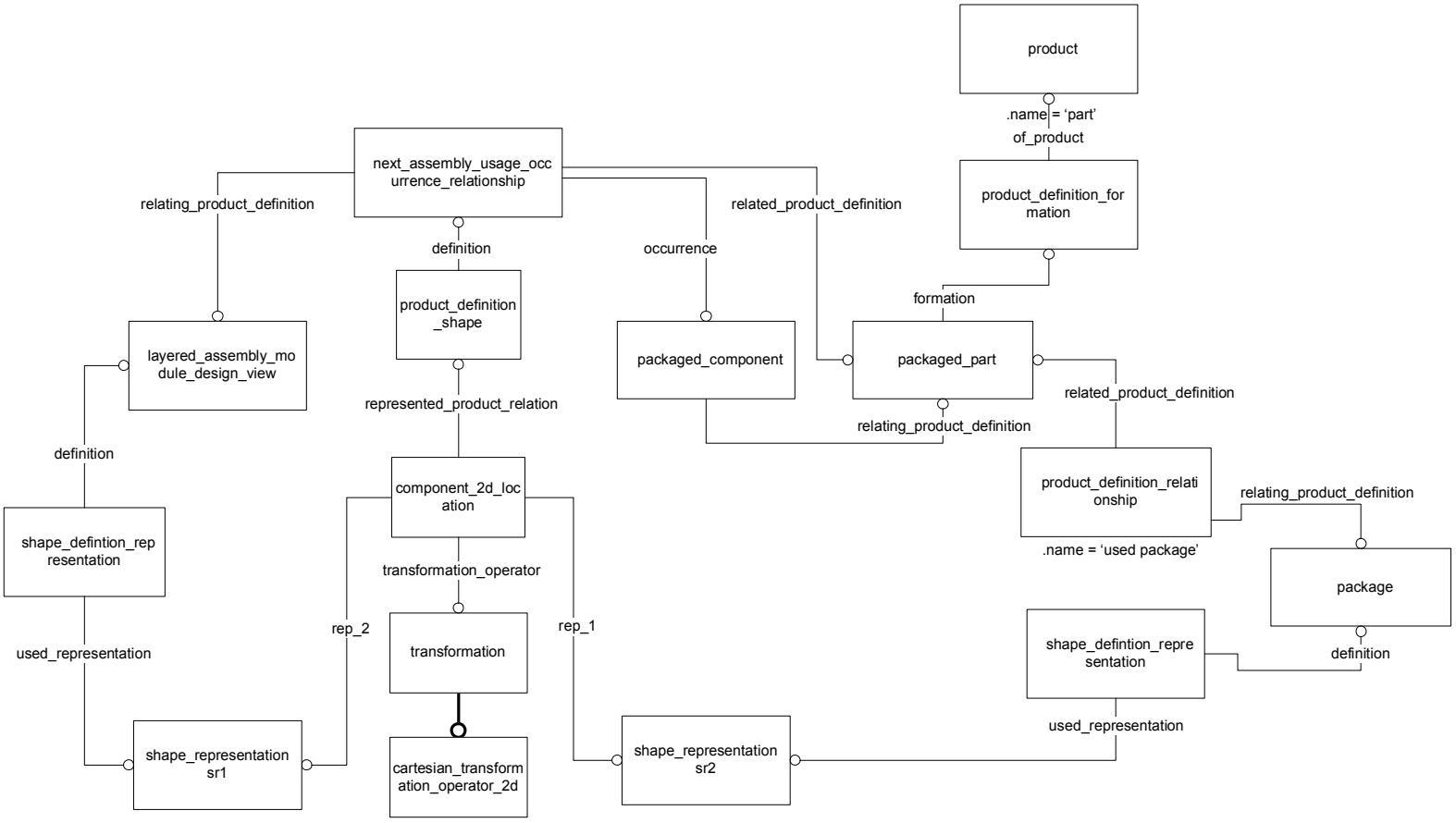
// Returns the product of which the given packaged_component is an instance.

product getProductOfPackagedComponent(packaged_component e_pc)
{
    packaged_part e_pp = referencedEntityOp(e_pc)
        where {e_pc.relating_product_definition->e_pp}

    product_definition_formation e_pdf = referencedEntityOp(e_pp)
        where {e_pp.formation->e_pdf}

    product e_p = referencedEntityOp(e_pdf)
        where {e_pdf.of_product->e_p}

    return e_p
}



// Returns the package that is used by a particular packaged_component. Note that there will often by multiple packages
// used by a particular packaged_part. In order to find the correct package, it is necessary to determine which shape_representation is
// located in the assembly through the component_2d_location and the Contextual_item_shape.

Package getPackageOfPackagedComponent(packaged_component pc, shape_representation sr1)
{
    Next_assembly_usage_occurrence_relationship nauor = referencingEntityOp(pc)
        where {nauor.occurrence->pc}

    Aggregate<product_definition_shape> a_pds = referencingEntitiesOp(nauor)
        where {product_definition_shape pds}
              {pds.definition->nauor}

    For Each product_definition_shape pds in a_pds
    {
        Aggregate<component_2d_location> a_c2dl = referencingEntitiesOp(pds)
            where {component_2d_location c2dl}
                  {c2dl.represented_product_relation->pds}

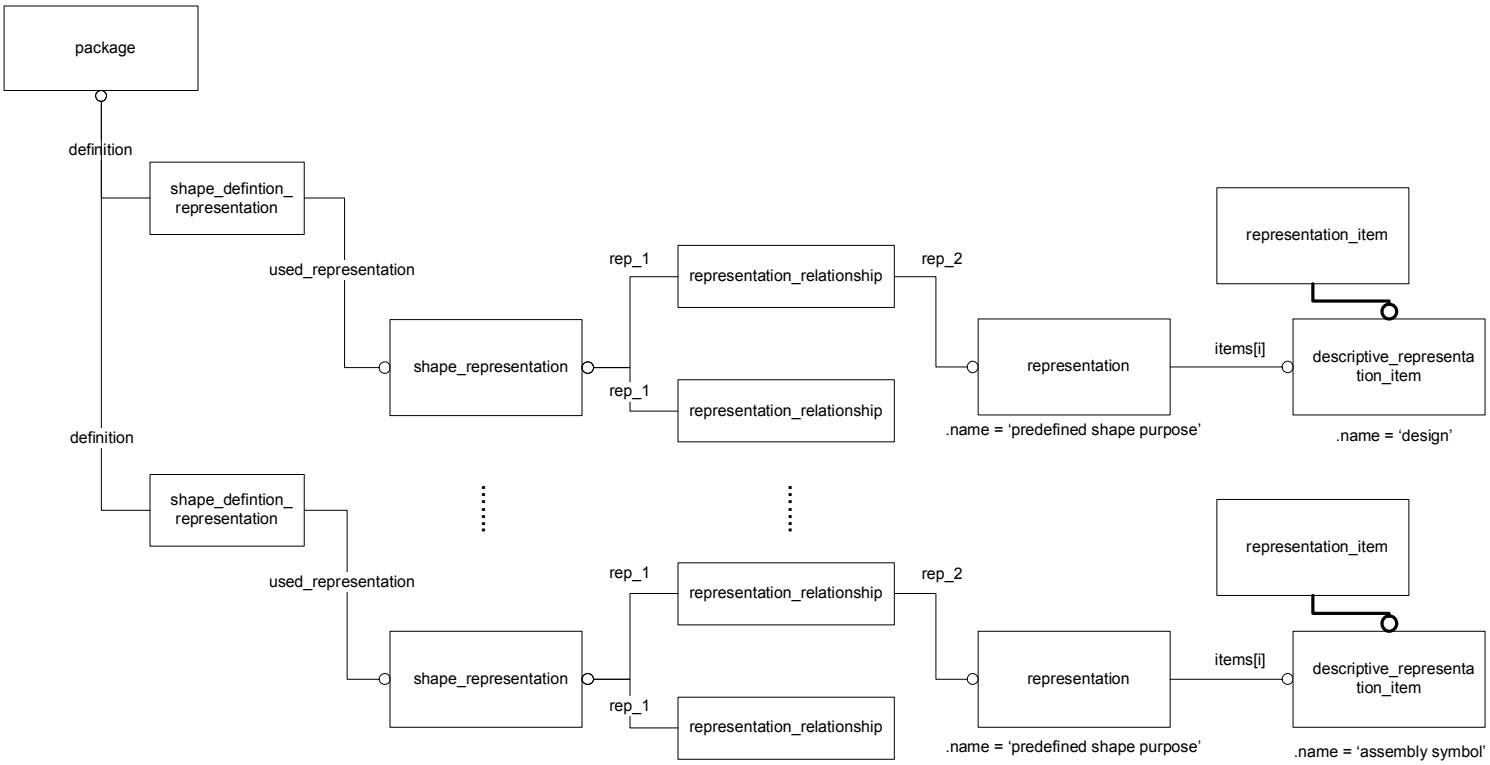
        component_2d_location e_c2dl = referencingFilterOp(a_c2dl, sr1)
            where {component_2d_cl e_c2dl member of a_c2dl}
                  {e_c2dl.rep_2 -> sr1}

        If (e_c2dl != null)
            Break For Each
    }

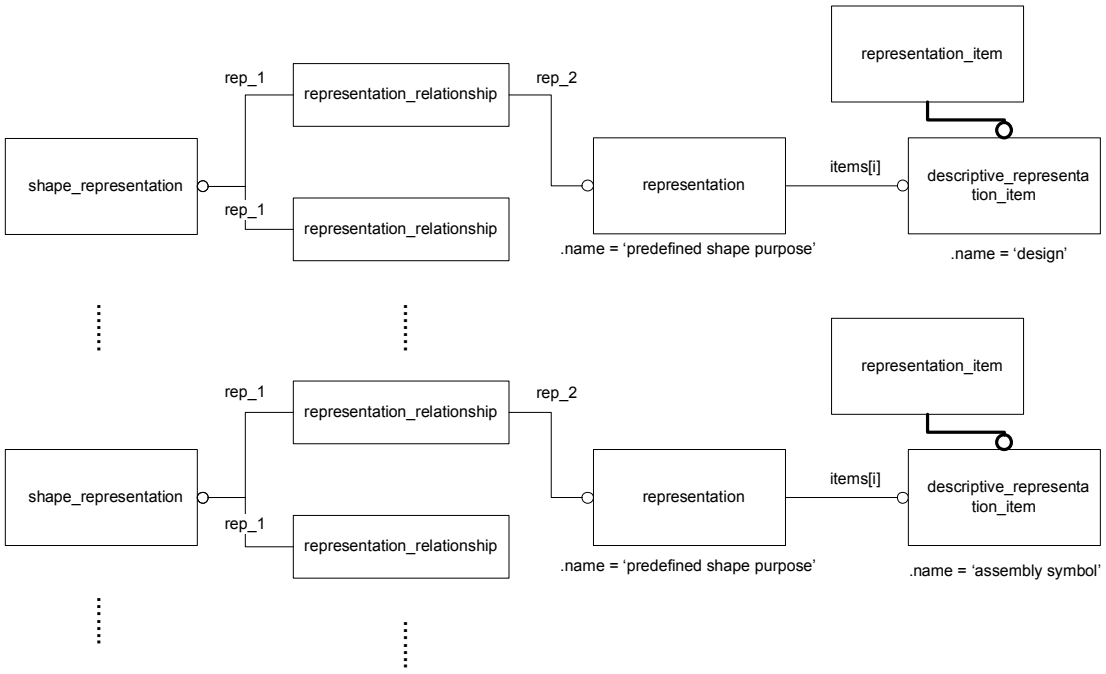
    shape_representation sr2 = referencedEntityOp(c2dl)
        where {c2dl.rep_1->sr2}

    package p = relatedEntityOp(sr2)
        where {shape_definition_representation sdr}
              {sr2<-sdr.used_representation}
              {sdr.defintion->p}

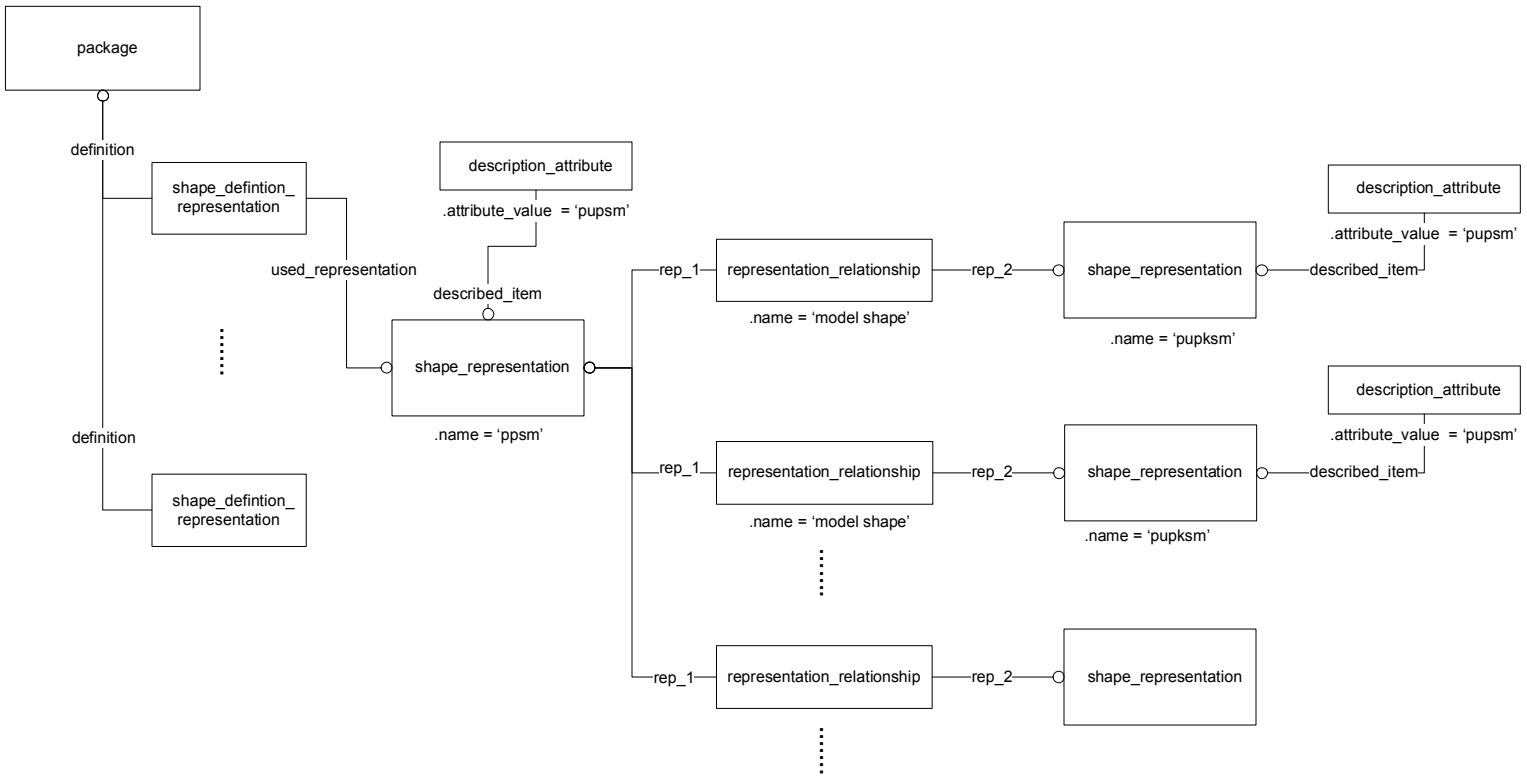
    return e_p;
}




```
shape_representation getShapeRepresentationOfPackageWithSpecifiedPurpose(package e_p, String purpose)
{
    Aggregate<shape_representation> a_sr = relatedEntitiesOp(e_p)
        where  {shape_representation e_sr}
               {shape_definition_representation e_sdr}
               {e_p <- e_sdr.definition}
               {e_sdr.used_representation -> e_sr}

    return getShapeRepresentationWithSpecifiedPurpose(a_sr, purpose);
}
```

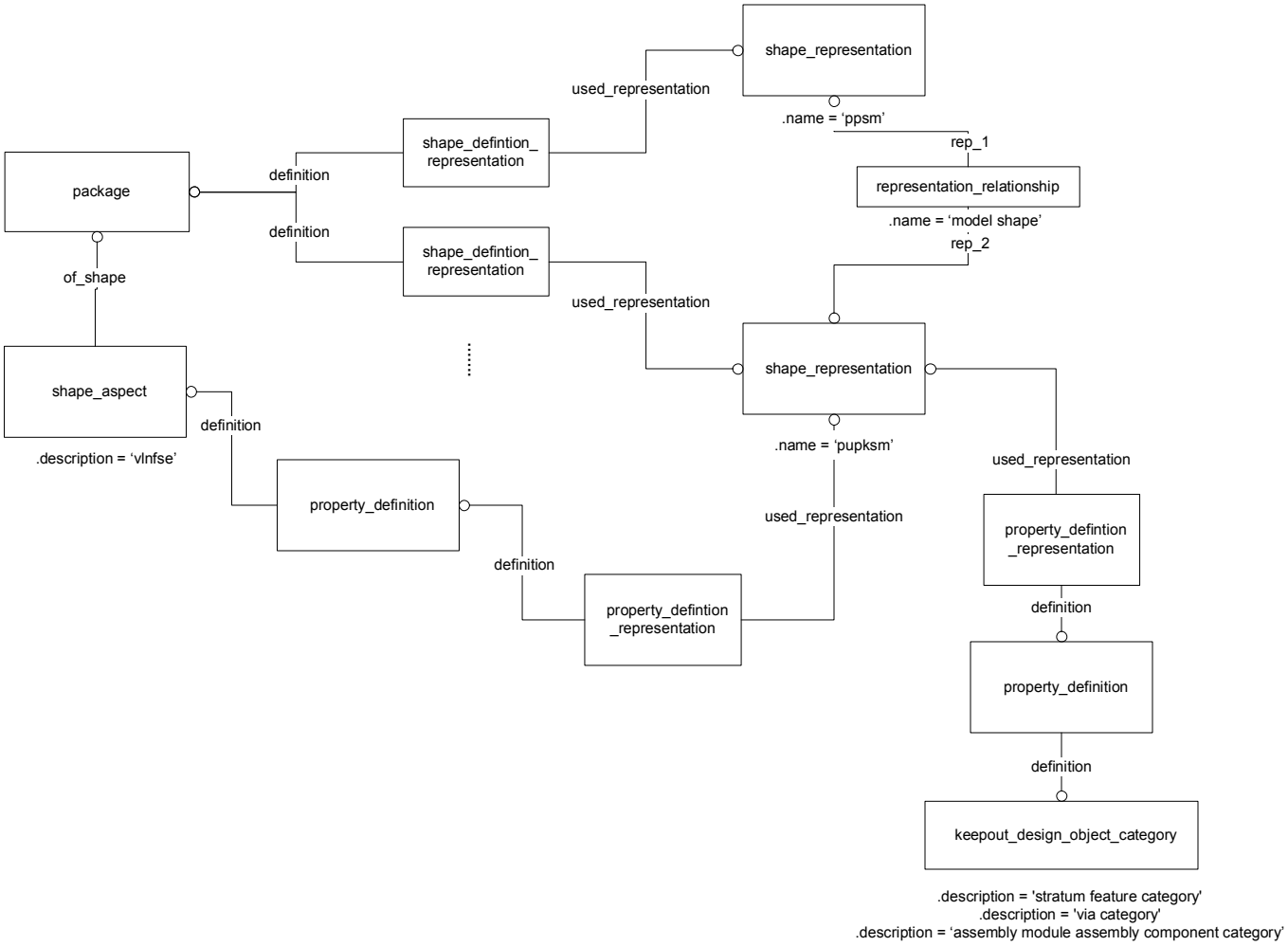



```
// Returns the shape_representation within the given set of shape_representations that has a specified 'predefined shape purpose.'
// If no such shape_representation exists, return null.

shape_representation getShapeRepresentationWithSpecifiedPurpose(Aggregate<shape_representation> a_sr, String purpose)
    {
        For each shape_representation e_sr in a_sr
        {
            Aggregate<representation> a_rep = relatedEntitiesOp(e_sr)
                where  {representation e_rep}
                       {representation_relationship e_rr}
                       {e_sr <- e_rr.rep_1}
                       {e_rr.rep_2 -> e_rep}

            For each representation e_rep in a_rep
            {
                if (e_rep.name == 'predefined shape purpose')
                {
                    Aggregate<representation_item> a_ri = ReferencedEntitiesOp(e_rep, purpose)
                        where  {representation_item e_ri}
                               {e_rep.items contains e_ri}
                               {e_ri.name = purpose}

                    if (not empty a_ri)
                        return e_sr
                }
            }
        }
        return null;
    }
```

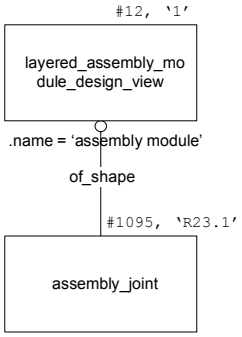


// Given a shape_representation corresponding to a mapping of ARM AO Physical_unit_planar_shape_model, this method will
// return all associated shape_representations corresponding to mappings of Physical_unit_planar_keepout_shape_model.

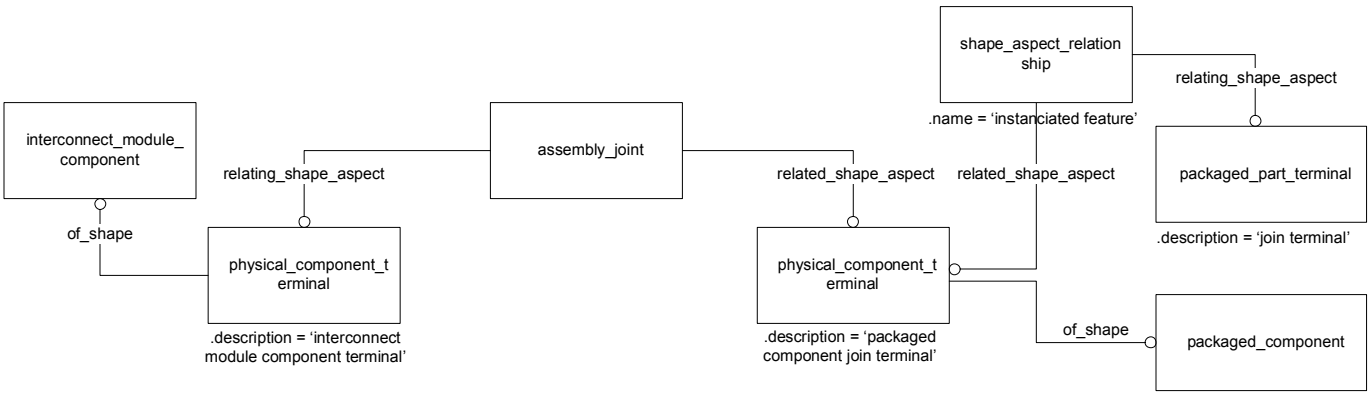
```
Aggregate<shape_representation> getAllKeepoutsForPhysicalUnitShapeModel(shape_representation e_pupsm)
    {
        Aggregate<shape_representation> a_sr = relatedEntitiesOp(e_pupsm, "model shape")
            where {shape_representation e_sr}
                  {representation_relationship e_rr}
                  {e_pupsm <- e_rr.rep_1}
                  {e_rr.name = 'model shape'}
                  {e_rr.rep_2 -> e_sr}

        AShape_representation a_keepouts = new AShape_representation();

        For each shape_representation e_sr in a_sr
        {
            if (e_sr.name = 'pupksm')
            {
                add e_sr to a_keepouts
            }
        }
        return a_keepouts
    }
```




```
// Given an aggregate of shape_representation corresponding to Physical_unit_planar_keepout_shape_model,
// Returns the first shape_representation whose associated keepout_design_object_category matches the given description

shape_representation getKeepoutShapeRepresentationWithSpecifiedKeepoutCategory
                                (Aggregate<shape_representation> a_sr, String givenDescription)
    {
        For each shape_representation e_sr in a_sr
        {
            Aggregate<property_definition> a_pd = relatedEntitiesOp(e_sr)
                where  {property_definition e_pd}
                       {property_definition_representation e_pdr}
                       {e_sr <- e_pdr.used_representation}
                       {e_pdr.definition -> e_pd}

            For each property_defintion e_pd in a_pd
            {
                entity e = e_pd.definition
                if (e is instance of keepout_design_object_category_
                {
                    keepout_design_object_category e_kdoc = e
                    if (e_kdoc.description == givenDescription)
                        return e_sr
                }
            }
        }
        return null;
    }
```




```
// Returns an aggregate of all assembly_joint in an layered_assembly_module_design_view.
// There is an assembly joint for each terminal of a packaged_part.

Aggregate<assembly_joint> getAllAssemblyJointsInPca(_assembly_module_design_view ad)
{
    Aggregate<assembly_joint> a_aj = referencingEntititesOp(ad)
        where  {assembly_joint aj}
                {aj.of_shape->ad}

    return  a_aj
}
```

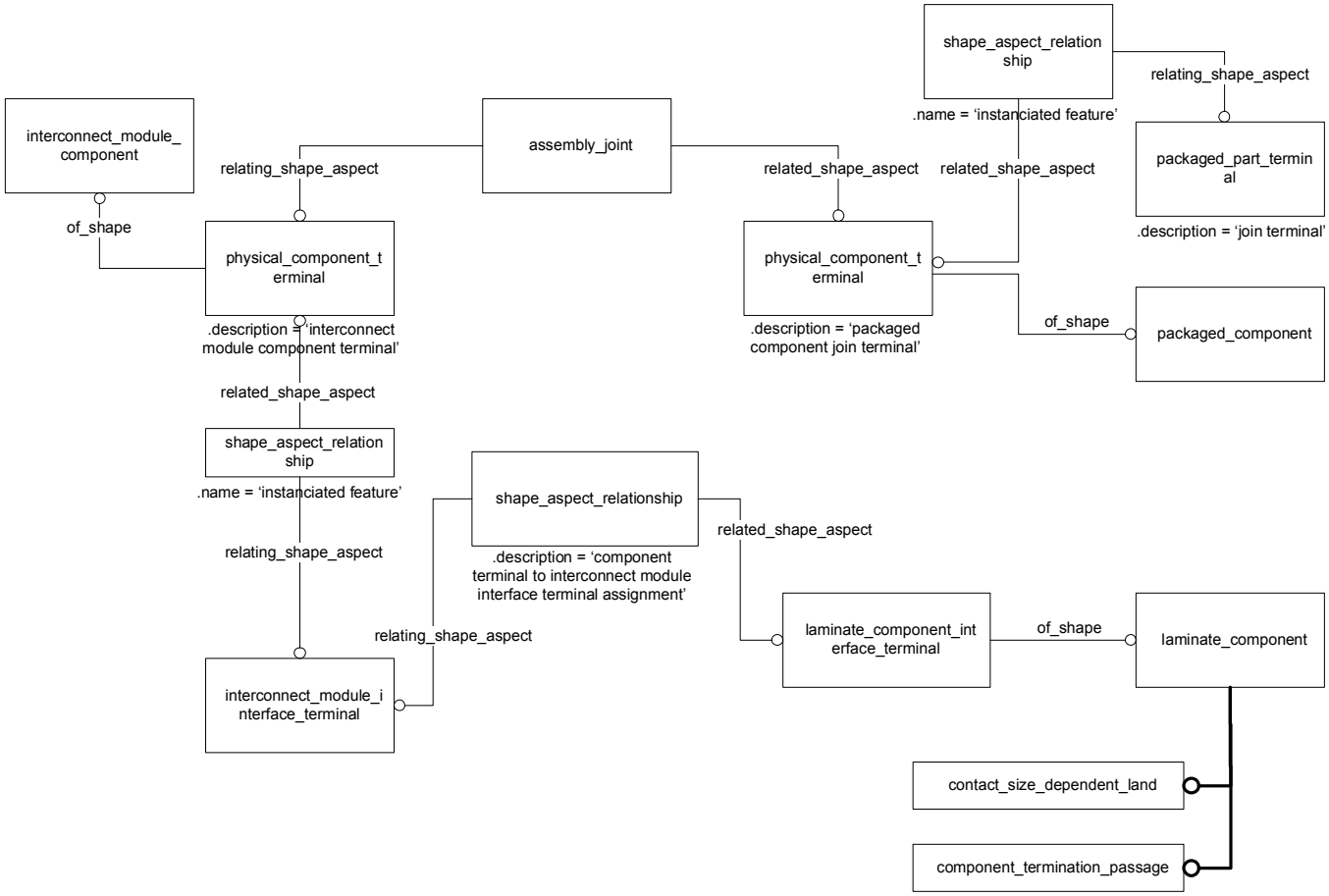


// Returns the packaged_part_terminal ('join terminal') for a given assembly_joint

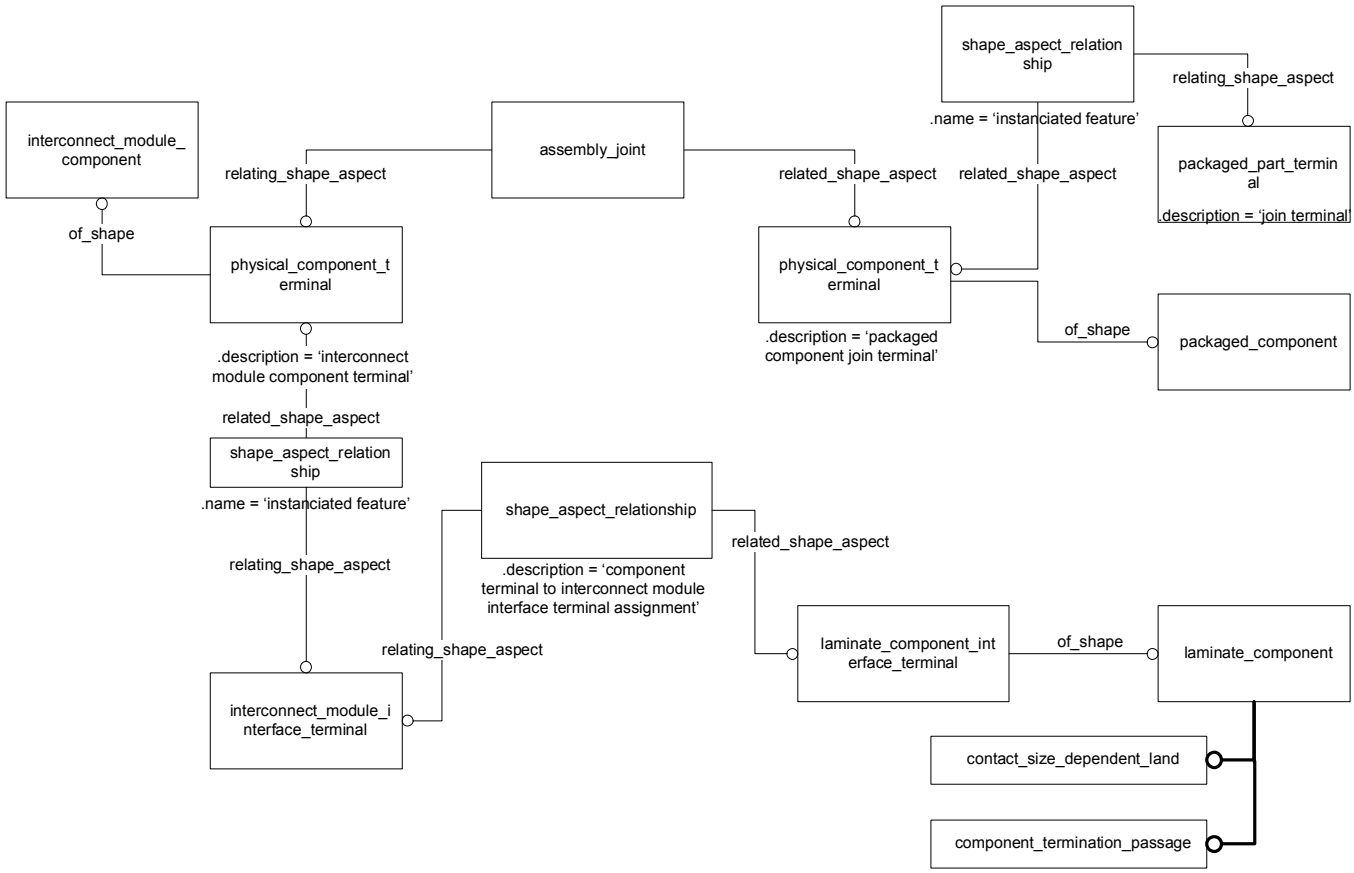
packaged_part_terminal getPackagedComponentAndJoinTerminalForAssemblyJoint(assembly_joint aj)
{
    physical_component_terminal pct = referencedEntityOp(aj)
        where {aj.related_shape_aspect->pct}
            {pct.description = 'packaged component join terminal'}

    packaged_part_terminal ppt = relatedEntityOp(pct)
        where {shape_aspect_relationship sar}
            {pct<-sar.related_shape_aspect}
            {pct.relating_shape_aspect->ppt}
            {sar.name = 'instanciated feature'}

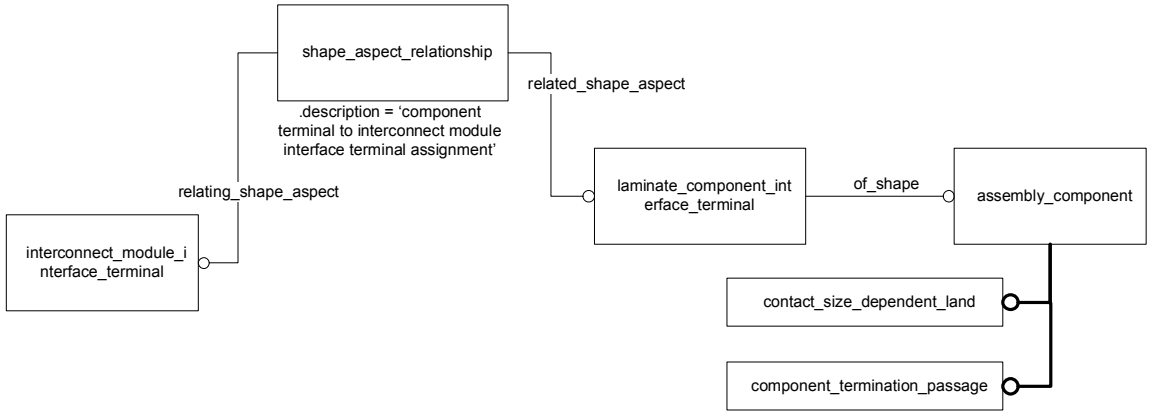
    if (ppt.description = 'join terminal')
        return ppt
    else
        return null
}



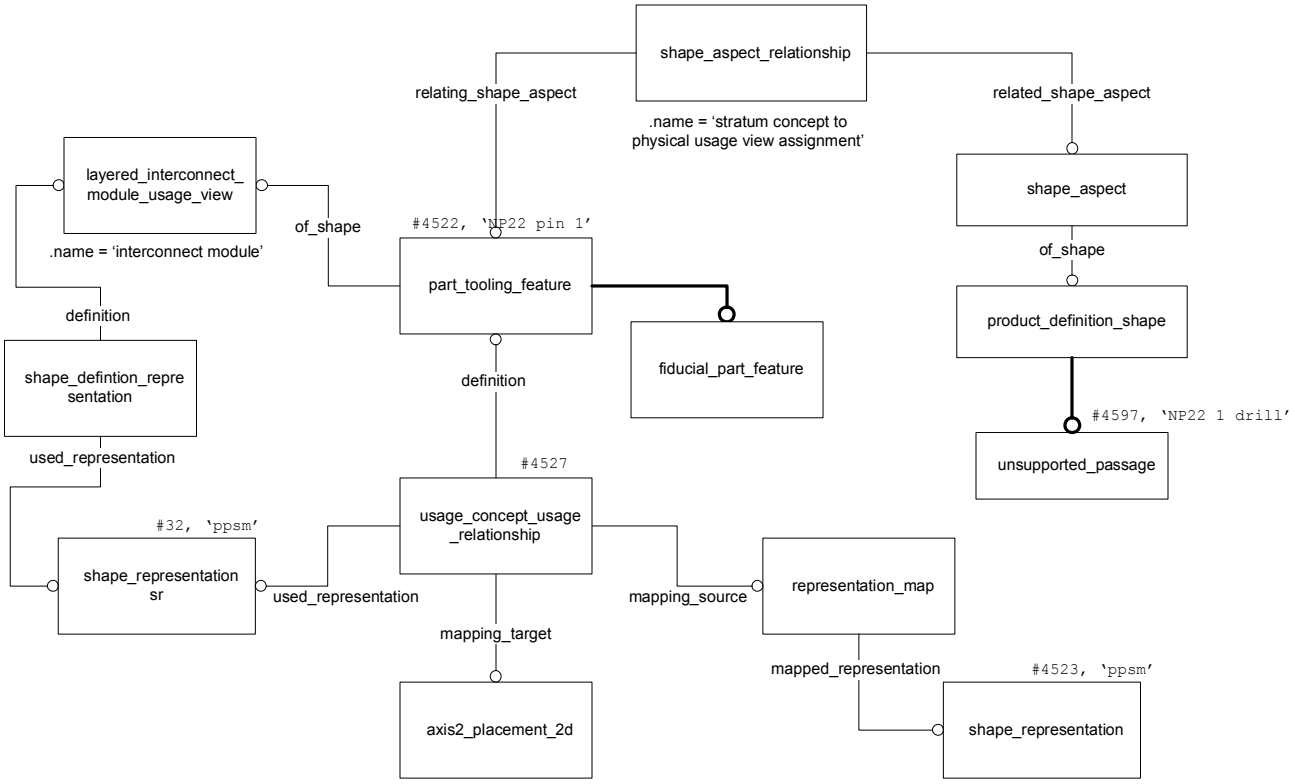

```
// Returns the associated laminate_component for a given assembly_joint. The returned entity is typically either a
// contact_size_dependent_land in the case of an assembly_joint for a surface joint terminal, or a
// component_termination_passage in the case of an assembly_joint for a through hole terminal.

laminate_component getLaminateComponentForAssemblyJoint(assembly_joint aj)
{
    physical_component_terminal imct = referencedEntityOp(aj)
        where {aj.relating_shape_aspect->imct}
              {pct.description = 'interconnect module component terminal'}

    interconnect_module_interface_terminal imit = relatedEntityOp(pct)
        where {shape_aspect_relationship sar}
              {pct<-sar.related_shape_aspect}
              {sar.relating_shape_aspect->imit}
              {sar.name = 'instanciated feature'}

    laminate_component_interface_terminal lcit = relatedEntityOp(imit)
        where {shape_aspect_relationship sar}
              {imit<-sar.relating_shape_aspect}
              {sar.related_shape_aspect->lcit}
              {sar.description = 'component terminal to interconnect module interface terminal assignment'}

    if (lcit.of_shape is instance of laminate_component)
    {
        laminate_component e_lc = lcit.of_shape
        return e_lc
    }
    return null
}
```




```
// Returns the associated laminate_component for a given interconnct_module_component_terminal.
// The returned entity is typically either a contact_size_dependent_land in the case of an assembly_joint for a surface joint terminal,
// or a component_termination_passage in the case of an assembly_joint for a through hole terminal.

laminate_component getLaminateComponentForIMCT(physical_component_terminal imct)
{
    interconnect_module_interface_terminal imit = relatedEntityOp(pct)
        where  {shape_aspect_relationship sar}
               {pct<-sar.related_shape_aspect}
               {sar.relating_shape_aspect->imit}
               {sar.name = 'instanciated feature'}

    laminate_component_interface_terminal lcit = relatedEntityOp(imit)
        where  {shape_aspect_relationship sar}
               {imit<-sar.relating_shape_aspect}
               {sar.related_shape_aspect->lcit}
               {sar.description = 'component terminal to interconnect module interface terminal assignment'}

    if (lcit.of_shape is instance of laminate_component)
    {
        laminate_component e_lc = lcit.of_shape
        return e_lc
    }
    return null
}
```



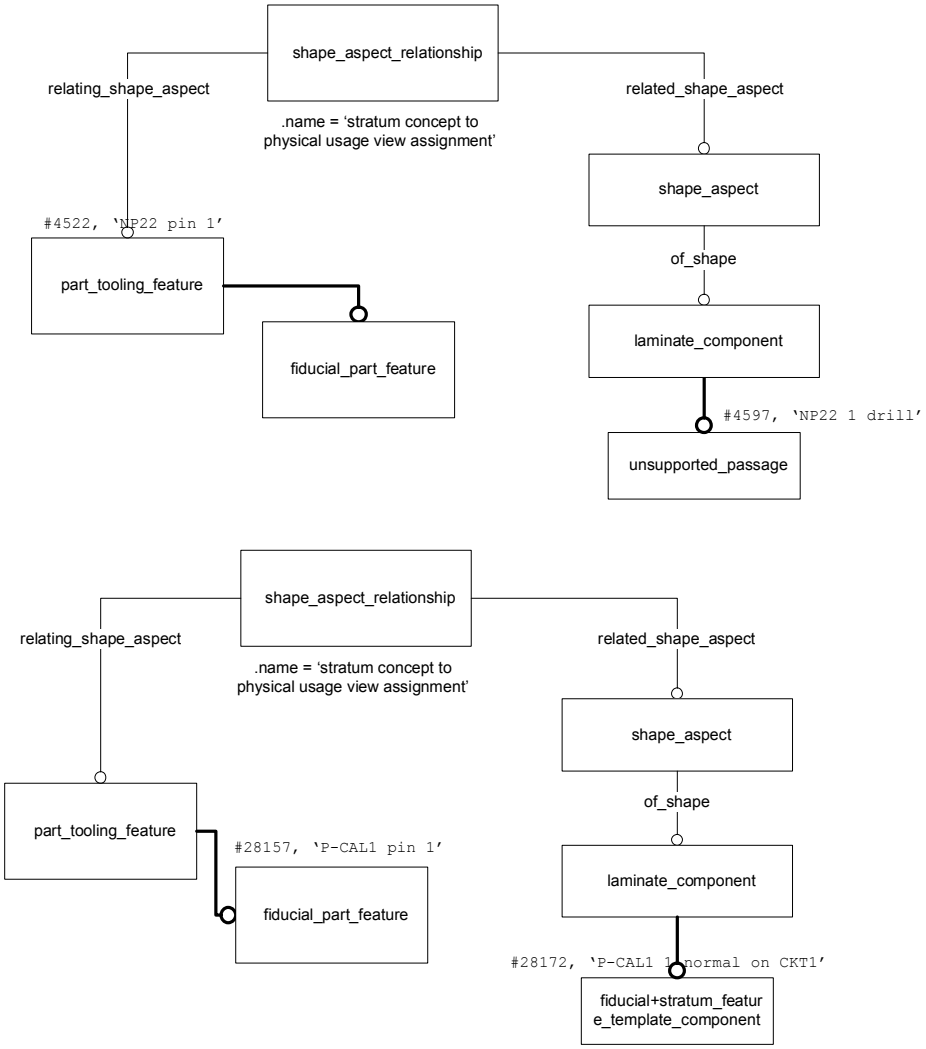

```
// Returns the associated imterconnect_module_interface_terminal for a given assembly_component if applicable.
// The given assembly_component may be a contact_size_dependent_land in the case of an assembly_joint for a
// surface joint terminal or a component_termination_passage in the case of an assembly_joint for a through hole terminal.

interconnect_module_interface_terminal getIMITforAC(assembly_component e_ac)
    {
        laminate_component_interface_terminal e_lcit = referencingEntityOp(e_ac)
            where  {e_ac <- e_lcit.of_shape}

        if (e_lcit == null)
            return null

    interconnect_module_interface_terminal e_imit = relatedEntityOp(e_lcit)
        where  {shape_aspect_relationship sar}
                {e_imit<-sar.relating_shape_aspect}
                {sar.related_shape_aspect->e_lcit}
                {sar.description = 'component terminal to interconnect module interface terminal assignment'}

    return e_imit
}
```



// Returns an aggregate of part_tooling_features that are located on the layered_interconnect_module_usage_view
// (associated with Pcb_usage_view).
// This includes part_tooling_features and its subtype fiducial_part_feature.
// Each of these Part_tooling_features are associated with a 'stratum concept' which is the mapping to an
// element of the Pcb.
// Note that the part_tooling_features are obtained through a usage_concept_usage_relationship with the
// shape_representation of the interconnect_definition (Pcb_usage_view).

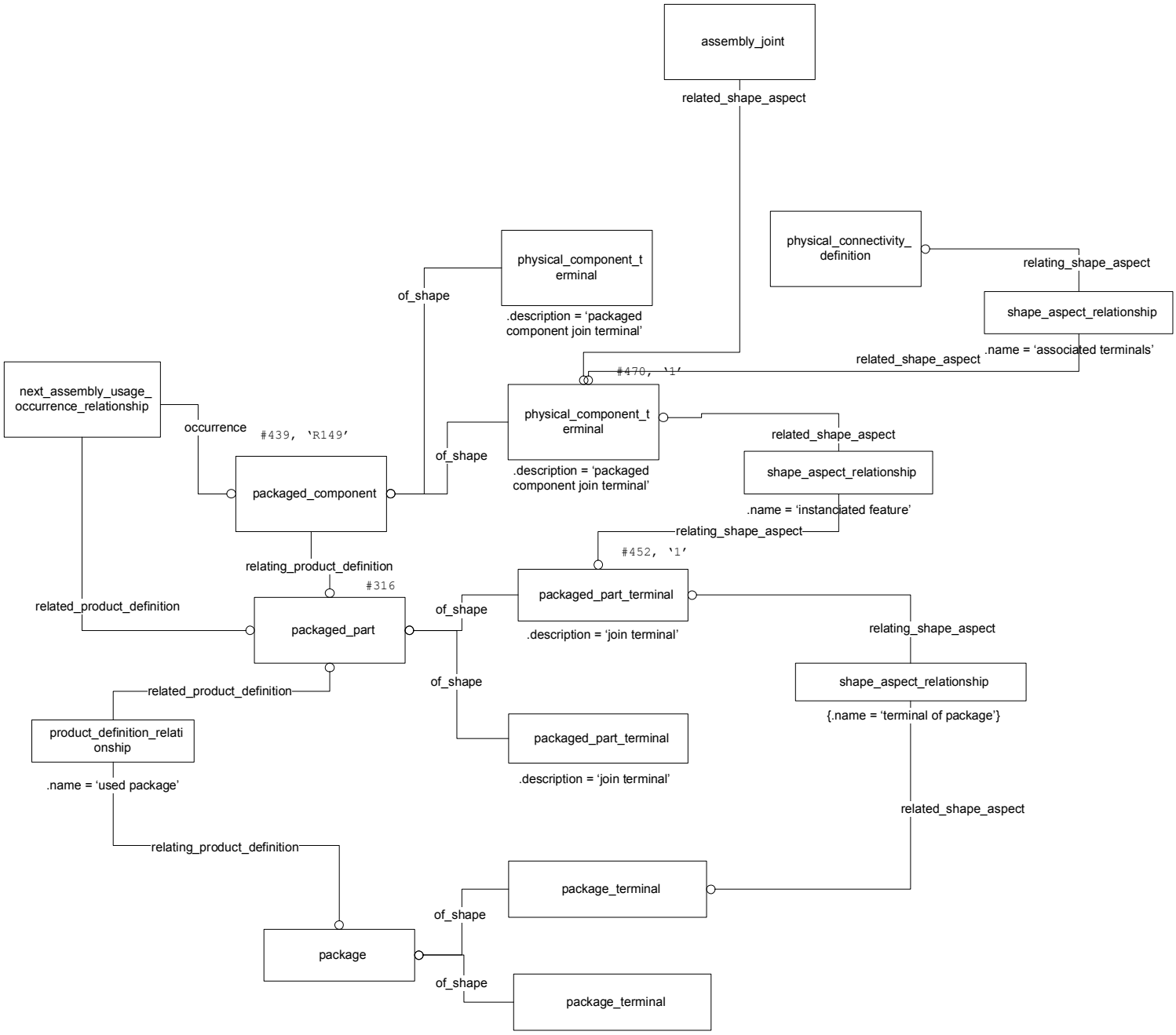
Aggregate<part_tooling_feature> getPartToolingFeaturesInPcb(layered_interconnect_module_usage_view id,
shape_representation sr)

```
{
    Aggregate<part_tooling_feature> a_ptf = relatedEntitiesOp(sr)
        where {part_tooling_feature ptf}
              {usage_concept_usage_relationship ucur}
              {sr<-ucur.used_representation}
              {ucur.definition->ptf}

    return  a_ptf
}
```



// Returns an associated laminate_component for a part_tooling_feature (or its subtype fiducial_part_feature).
// Examples include an unsupported_passage (a tooling hole) in the case of a part_tooling_feature or a
// fiducial+stratum_feature_template_component in the case of a fiducial_part_feature.
// In the event that there is not an associated laminate_component, the query returns null.

```
laminate_component getLaminateComponentForPartToolingFeature(Part_tooling_feature ptf)
{
    shape_aspect sa = relatedEntityOp(ptf)
        where {shape_aspect_relationship sar}
              {ptf<-sar.relating_shape_aspect}
              {sar.related_shape_aspect->sa}
              {sar.name = 'stratum concept to physical usage view assignment'}

    laminate_component lc = referencedEntityOp(sa)
        where {sa.of_shape->lc}

    return lc
}
```


assembly_joint
related_shape_aspect
physical_component_terminal
.description = 'packaged component join terminal'
of_shape
physical_connectivity_definition
relating_shape_aspect
shape_aspect_relationship
.name = 'associated terminals'
related_shape_aspect
#470, '1'
next_assembly_usage_occurrence_relationship
occurrence
#439, 'R149'
physical_component_terminal
.description = 'packaged component join terminal'
of_shape
related_shape_aspect
shape_aspect_relationship
.name = 'instanciated feature'
relating_shape_aspect
#452, '1'
packaged_component
relating_product_definition
#316
related_product_definition
packaged_part
packaged_part_terminal
.description = 'join terminal'
of_shape
relating_shape_aspect
shape_aspect_relationship
{.name = 'terminal of package'}
of_shape
related_product_definition
product_definition_relationship
.name = 'used package'
packaged_part_terminal
.description = 'join terminal'
related_shape_aspect
relating_product_definition
package_terminal
of_shape
package
of_shape
package_terminal